

UNIVERSITÉ DE MONTRÉAL

RÉALISATION D'UN SYSTÈME DE COMMUNICATIONS SANS-FIL SUR UNE  
PLATEFORME DE PROTOTYPAGE RAPIDE

YONY KEVIN AH-KIOON  
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLOME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(GÉNIE ÉLECTRIQUE)  
JUN 2009



Library and Archives  
Canada

Published Heritage  
Branch

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque et  
Archives Canada

Direction du  
Patrimoine de l'édition

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file Votre référence*  
ISBN: 978-0-494-57239-9  
*Our file Notre référence*  
ISBN: 978-0-494-57239-9

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

---

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

RÉALISATION D'UN SYSTÈME DE COMMUNICATIONS SANS-FIL SUR UNE  
PLATEFORME DE PROTOTYPAGE RAPIDE

présenté par: AH-KIOON Yony Kevin

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. CARDINAL Christian, Ph.D., président

M. FRIGON Jean-François, Ph.D., membre et directeur de recherche

M. DAVID Jean-Pierre, Ph.D., membre

À ceux qui m'ont aidé.



## REMERCIEMENTS

Je tiens à exprimer ma reconnaissance envers mon directeur de recherche, Dr. Jean-François Frigon, pour sa collaboration, ses précieux conseils et ses commentaires dans la réalisation de ce travail.

Je remercie aussi mes collègues pour leur contribution au niveau de suggestions et d'échanges d'idées dans le développement de ce projet.

J'aimerais exprimer ma gratitude envers mon père, ma mère, Steve, Keith et Cindy pour leur présence et leur encouragement. Un merci spécial à mon frère pour me fournir ses conseils et me faire part de ses expériences de recherche.

Je remercie Marie-France Nguyen et sa famille pour leur support au courant de mes années d'études au Canada.

Je remercie Quan Yim pour m'aider à garder patience et calme dans les moments les plus difficiles.

Finalement, j'aimerais remercier les membres du comité du jury, Dr. Christian Cardinal et Dr. Jean-Pierre David pour prendre le temps de lire et de commenter ce mémoire.

## RÉSUMÉ

L'industrie des télécommunications sans-fil connaît une grande croissance depuis quelques années. Plusieurs nouveaux produits sont proposés aux consommateurs chaque année. La décision de permettre l'utilisation non-licenciée de certaines bandes de fréquences a aussi favorisé le développement d'une multitude d'équipements sans-fil personnels, tels que les réseaux locaux sans-fil (WLAN) à courte portée ou encore les écouteurs sans-fil. Parallèlement, les avancées dans les techniques de prototypage rapide ont contribué à faciliter la validation et l'évaluation de divers algorithmes de communications en augmentant la performance des composantes électroniques utilisées et en réduisant le coût associé. Dans le cadre de ce mémoire, nous faisons la conception théorique et matérielle d'un système de communications sans-fil à bande étroite. Le projet décrit dans ce document est divisé en deux parties.

Dans la première partie, un système complet de communications est développé et simulé à l'aide de l'outil de simulation Simulink. Le canal considéré est à évanouissement lent et non-sélectif en fréquence. Grâce à un contrôleur de gain automatique (AGC), l'amplitude du signal reçu est maintenue dans une plage permettant une utilisation efficace des convertisseurs analogique-numérique. Un algorithme de détection de trames est utilisé pour détecter l'arrivée des paquets et déclencher les boucles de synchronisation. La phase et l'amplitude du signal reçu sont corrigées avec un filtre adaptatif de type *Least Mean Square* (LMS). L'algorithme de recouvrement d'horloge utilise les transitions entre symboles pour estimer le temps d'échantillonnage idéal. La boucle PLL requise pour la synchronisation de fréquence est basée sur un algorithme utilisant l'erreur de phase entre le symbole reçu et le point de la constellation le plus proche.

Ainsi, ces différents algorithmes sont tout d'abord vérifiés et validés à l'aide de l'outil Simulink. La performance est évaluée en vérifiant le taux d'erreurs binaires pour différents niveaux de rapport signal à bruit (SNR). En comparant la performance simulée avec celle d'un système idéal, nous observons une dégradation maximale de 0.3 dB pour des SNR allant de 8 dB à 13 dB.

Par la suite, ce système est implémenté en partie sur une plateforme de prototypage rapide de la compagnie Lyrtech Inc. L'implémentation effectuée constitue la partie bande de base du système de communications. Elle permet d'obtenir une idée de la faisabilité et de la performance du système. Le transmetteur et l'algorithme de recouvrement d'horloge ainsi que le filtre LMS et le bloc de détection de trame ont été programmés sur la plateforme. La calibration des paramètres utilisés est faite de façon heuristique. En général, nous avons utilisé les valeurs obtenues dans la partie simulation comme point de départ. Les résultats obtenus montrent une perte de 0.8 dB par rapport à la courbe théorique et de 0.5 dB par rapport à la courbe obtenue par simulation. Des améliorations pouvant être apportées au système sont finalement proposées.

## ABSTRACT

The telecommunications industry is growing fast with several new services being proposed to customers every year. Unlicensed use of part of the frequency spectrum has certainly helped in the development of various additional personal communications services, ranging from home wireless Local Area Networks (WLAN) to wireless headsets. At the same time, hardware prototyping and implementation are increasingly converging towards a reasonably affordable reach as well as an improved performance. In this thesis, we present the development of a broadband wireless communications system. The document is mainly divided in two sections.

In the first part, a full broadband communications system is developed and simulated using Simulink for a slow fading non-selective channel. An automatic gain control (AGC) loop is used for power control at the input of the receiver in order to ensure efficient utilization of the dynamic range of analog to digital converters. A frame detection algorithm ensures the detection of incoming packets so as to trigger synchronization loops. The phase and the amplitude errors of the received signal are fine tuned with a Least Mean Square (LMS) filter. In addition to this, we have identified timing and frequency inconsistencies as the main potential sources of error. An algorithm based on the sign of the half-baud symbol is used for timing recovery. Frequency synchronization is accomplished using an algorithm based on the sign of the imaginary part of the projection of the incoming data symbols on their respective closest constellation points.

These different algorithms are implemented and simulated for validation. Finally, the performance of the system as a whole is evaluated through the Bit Error Rate (BER) for different levels of Signal to Noise Ratio (SNR). Comparison of the achieved performance shows that a maximum 0.3 dB power loss compared to Shannon's theoretical limit is observed for a signal to noise ratio range from 8 dB to 13 dB.

This system is meant to be implemented on a rapid prototyping platform solution proposed by Lyrtech Inc. Part of the baseband implementation of the communications system follows in order to give a general idea of the feasibility of the project as well as the perfor-

mance characteristics that can be expected from the communications systems. The clock synchronization algorithm together with the LMS filter and frame detection block are integrated. Their parameters are heuristically calibrated, usually starting with initial values obtained from simulation parameters. Results show that there is a 0.8 dB power loss compared to Shannon's theoretical limit and a 0.5 dB power loss compared to simulated results. Finally, potential improvement and future works are proposed.

## TABLE DES MATIÈRES

DÉDICACE . . . . .	iv
REMERCIEMENTS . . . . .	v
RÉSUMÉ . . . . .	vi
ABSTRACT . . . . .	viii
TABLE DES MATIÈRES . . . . .	x
LISTE DES FIGURES . . . . .	xii
LISTE DES ACRONYMES . . . . .	xiii
LISTE DES TABLEAUX . . . . .	xv
LISTE DES ANNEXES . . . . .	xvi
INTRODUCTION . . . . .	1
CHAPITRE 1      CONCEPTS DE BASE . . . . .	5
1.1    Le canal sans-fil . . . . .	5
1.1.1    Bruit additif . . . . .	6
1.1.2    Atténuation . . . . .	8
1.1.3    Multi-trajets . . . . .	9
1.1.4    Interférences . . . . .	12
1.2    La modulation . . . . .	13
1.3    La conversion en bande passante . . . . .	14
1.4    La période d'un symbole . . . . .	16
1.5    Détection optimale . . . . .	17
1.6    L'horloge . . . . .	18

1.7	Oscillateurs . . . . .	20
1.8	Système global . . . . .	21
CHAPITRE 2 SURVOL DES TECHNIQUES DE PROTOTYPAGE DE SYS-		
	TÈMES DE COMMUNICATIONS . . . . .	23
2.1	Critères de développement . . . . .	23
2.2	La radio logicielle . . . . .	24
2.3	Traitement hors ligne . . . . .	25
2.4	DSP . . . . .	26
2.5	ASIC . . . . .	27
2.6	FPGA . . . . .	28
2.7	Prototypes hybrides . . . . .	29
CHAPITRE 3 LES ÉLÉMENTS DU MODÈLE DE SIMULATION . . . . .		
3.1	Format des paquets transmis . . . . .	31
3.1.1	Séquence de début des trames . . . . .	33
3.1.2	Séquence de synchronisation . . . . .	34
3.1.3	Séquence de fin des trames . . . . .	35
3.2	Le modulateur et le démodulateur . . . . .	35
3.3	Filtres de Nyquist . . . . .	36
3.4	Filtre d'égalisation . . . . .	41
3.5	Circuit d'ajustement initial de puissance . . . . .	45
3.5.1	Circuit utilisé . . . . .	45
3.5.2	Tests de performance . . . . .	48
3.6	Détection de début de trame de synchronisation . . . . .	48
3.7	Synchronisation du récepteur . . . . .	51
3.7.1	Synchronisation de phase et de fréquence . . . . .	53
3.7.2	Recouvrement d'horloge . . . . .	56
3.8	Simulations et résultats . . . . .	62

CHAPITRE 4	LES APPAREILS DE PROTOTYPAGE . . . . .	65
4.1	Vue d'ensemble du matériel . . . . .	65
4.2	Description hardware du Signal Master QUAD . . . . .	67
4.3	Description hardware du VHS-ADC . . . . .	75
4.4	Description software . . . . .	76
4.5	Outils nécessaires au développement . . . . .	82
CHAPITRE 5	RÉALISATION ET PERFORMANCE . . . . .	86
5.1	Le transmetteur . . . . .	86
5.2	Le récepteur . . . . .	90
5.2.1	Circuit utilisé au VHS-ADC . . . . .	94
5.2.2	Détection de début de paquet . . . . .	94
5.2.3	Algorithme de recouvrement d'horloge . . . . .	95
5.2.4	Filtre d'égalisation . . . . .	99
5.2.5	Enregistrement des données reçues . . . . .	100
5.2.6	Blocs DSP . . . . .	103
5.3	Performances . . . . .	104
5.3.1	Caractérisation du transmetteur . . . . .	105
5.3.2	Vérification de la fonctionnalité du récepteur . . . . .	109
5.3.3	Performance du récepteur . . . . .	109
CHAPITRE 6	DISCUSSIONS ET CONCLUSION . . . . .	116
6.1	Conclusion . . . . .	116
6.2	Améliorations . . . . .	116
6.3	Perspectives . . . . .	118
RÉFÉRENCES	. . . . .	120
ANNEXES	. . . . .	125



## LISTE DES FIGURES

FIGURE 1.1	Température du bruit pour un système en cascade . . . . .	6
FIGURE 1.2	Les causes de multi-trajectoires . . . . .	10
FIGURE 1.3	Conversion de bande de base à bande passante au transmetteur et de bande passante à bande de base au récepteur . . . . .	16
FIGURE 1.4	Structure d'un récepteur optimal . . . . .	17
FIGURE 1.5	Discrétisation du filtre adapté . . . . .	18
FIGURE 1.6	Effet du délai de canal sur le moment d'échantillonnage . . . . .	19
FIGURE 1.7	Une vue de haut niveau du système de communications . . . . .	22
FIGURE 2.1	Plateforme pour un SDR sur DSP [Stuber G.L. et al, 2004] . . . . .	27
FIGURE 2.2	Modèle d'un système FPGA [Barbero, L.G. & Thompson, J.S., 2005] . . . . .	28
FIGURE 3.1	Schéma bloc du transmetteur . . . . .	31
FIGURE 3.2	Structure du récepteur modélisé . . . . .	32
FIGURE 3.3	Vue générale des paquets . . . . .	33
FIGURE 3.4	Génération d'un PBRS avec un registre à décalage. . . . .	34
FIGURE 3.5	Interférences entre symboles. . . . .	36
FIGURE 3.6	Réponses des filtres en cosinus surélevés. . . . .	38
FIGURE 3.7	Réponse impulsionnelle du filtre en racine cosinus surélevé pour mise en forme d'impulsions. . . . .	40
FIGURE 3.8	Symboles à la sortie du filtre adapté. . . . .	41
FIGURE 3.9	Filtre LMS à 1 tap. . . . .	43
FIGURE 3.10	Taux de convergence de l'algorithme LMS pour différentes va- leurs de $\mu$ . . . . .	44
FIGURE 3.11	Contrôleur de gain automatique. . . . .	47
FIGURE 3.12	Résultat des tests effectués sur le contrôleur de puissance. . . . .	49
FIGURE 3.13	Circuit de détection de début de trame. . . . .	50
FIGURE 3.14	Circuit de détection de séquence de Barker. . . . .	51

FIGURE 3.15	Taux de pertes de paquets pour différents niveaux de SNR avec un seuil de détection à 6 . . . . .	52
FIGURE 3.16	Schéma bloc de la boucle de correction de phase. . . . .	53
FIGURE 3.17	Comparaison de phases du signal de réception et de la référence dans le plan I-Q . . . . .	55
FIGURE 3.18	Estimation d'erreur de phase. . . . .	55
FIGURE 3.19	Circuit d'un DDFS. . . . .	56
FIGURE 3.20	Illustration simplifiée des transitions possibles lors de la transmission d'un signal BPSK. . . . .	58
FIGURE 3.21	Circuit de recouvrement d'horloge . . . . .	59
FIGURE 3.22	Exemple de synchronisation d'horloge . . . . .	60
FIGURE 3.23	Effet d'un décalage d'un demi échantillon . . . . .	62
FIGURE 3.24	Résultat de simulation sur Simulink . . . . .	64
FIGURE 4.1	Photos des plateformes de développement . . . . .	66
FIGURE 4.2	Composantes du SMQUAD et leurs interconnexions . . . . .	68
FIGURE 4.3	Les blocs FastBus . . . . .	69
FIGURE 4.4	Interface Simulink du bloc RapidChannel. . . . .	72
FIGURE 4.5	Les blocs utilisés pour l'enregistrement de données . . . . .	74
FIGURE 4.6	Diagramme bloc des principales composantes du VHS-ADC . . . . .	75
FIGURE 4.7	Les convertisseurs et leurs paramètres . . . . .	77
FIGURE 4.8	Blocs des Registres du VHS-ADC . . . . .	78
FIGURE 4.9	Le SMCCE comme un pont de communications . . . . .	80
FIGURE 4.10	Interface Graphique du Record and Playback Utility . . . . .	81
FIGURE 4.11	Lecture des données enregistrées à partir d'un fichier binaire . . . . .	83
FIGURE 4.12	Étapes de développement [Lyrtech SMQ DesignGuide]. . . . .	84
FIGURE 5.1	Diagramme bloc du transmetteur . . . . .	86
FIGURE 5.2	Représentation et interprétation de chaque symbole . . . . .	87
FIGURE 5.3	Circuit utilisé pour générer les paquets . . . . .	88

FIGURE 5.4	Interface utilisateur pour la connexion . . . . .	90
FIGURE 5.5	Architecture du récepteur . . . . .	91
FIGURE 5.6	Modèle Simulink du FPGA du récepteur sur le VHS-ADC . . . . .	92
FIGURE 5.7	Modèle Simulink du FPGA du récepteur sur le SMQuad . . . . .	93
FIGURE 5.8	Diagramme temporel des états de RecEn et de SyncEn . . . . .	95
FIGURE 5.9	Schéma bloc du circuit de recouvrement d'horloge . . . . .	95
FIGURE 5.10	Détail bloc des circuits de recouvrement d'horloge . . . . .	96
FIGURE 5.11	Diagramme temporel de la génération d'horloge . . . . .	98
FIGURE 5.12	Dynamique de l'échantillonnage . . . . .	99
FIGURE 5.13	Circuit de multiplication complexe . . . . .	101
FIGURE 5.14	Circuit d'ajustement du gain du filtre LMS . . . . .	102
FIGURE 5.15	Logique de sélection des données enregistrées . . . . .	103
FIGURE 5.16	Blocs utilisés au DSP du récepteur . . . . .	104
FIGURE 5.17	Circuit pour la caractérisation du transmetteur . . . . .	105
FIGURE 5.18	Diagrammes d'oeil pour une boucle interne . . . . .	106
FIGURE 5.19	Diagrammes d'oeil pour une boucle externe . . . . .	108
FIGURE 5.20	Constellations des signaux de réception dans le cas d'une transmission en boucle . . . . .	110
FIGURE 5.21	Architecture utilisée pour la vérification du récepteur . . . . .	111
FIGURE 5.22	Courbe de BER vs SNR . . . . .	114
FIGURE I.1	Récepteur à corrélation . . . . .	126
FIGURE I.2	Structure d'un récepteur pour la détection d'un signal dans le bruit AWGN . . . . .	127
FIGURE I.3	Fonctions de densité de probabilité conditionnelles dans un canal AWGN . . . . .	129
FIGURE I.4	Constellation QPSK . . . . .	130
FIGURE I.5	Estimation linéaire d'une source à partir d'un signal de réception .	132
FIGURE I.6	Nombre en format binaire . . . . .	134

FIGURE II.1	DSP de l'émetteur . . . . .	137
FIGURE II.2	FPGA sur le SMQuad au transmetteur . . . . .	142
FIGURE II.3	FPGA sur le VHS du transmetteur . . . . .	145

## LISTE DES ACRONYMES

### ABBREVIATIONS

ADC :	Convertisseur analogique-numérique
AGC :	Contrôleur de Gain Automatique
AWGN :	Bruit Blanc Gaussien Additif
BER :	Taux d'erreurs binaires
CLK :	Signal d'horloge
$CLK_{Sample}$ :	Signal d'horloge d'un échantillon
DAC :	Convertisseur numérique à analogique
EMIF :	<i>External Mode Interface</i>
ISI :	Interférence Entre Symboles
LPF :	Filtre Passe-bas
MB :	MégaOctets
pdf :	Fonction de Densité de probabilité
PLL :	Phase-Locked Loop
PRBS :	Séquence binaire pseudo-aléatoire
QPSK :	Quadrature Phase Shift Keying
SDRAM :	Mémoire Vive
SMQUAD :	Plateforme Signal Master Quad
SNR :	Rapport Signal à bruit

### FONCTIONS DÉFINIES

$e^x$ :	Exponentiel
$\log(x)$ :	Logarithm base 10
$Q(x)$ :	Intégrale de la fonction de densité de probabilité Gaussienne de $x$ à $\infty$
$\Re$ :	Partie réelle
$\Im$ :	Partie imaginaire
$z^{-N}$ :	Total de N délais

**INDICES**

$dB :$	Décibels
$h :$	Constante de Plank ( $6.625 \times 10^{-34} J - s$ )
$k :$	Contante de Boltzmann ( $1.38 \times 10^{-23} J/K$ )
$T :$	Température (K)
$T :$	Période
$\phi :$	Angle ou phase
$f :$	Fréquence (Hz)
framesize :	Longueur d'une trame (Nombre de symboles)
G, K :	Gain d'amplification
L :	Taux de suréchantillonnage
P :	Puissance
$\Gamma :$	Coefficient de réflexion de Fresnel
$N_0 :$	Puissance de bruit
$r(t) :$	Signal de réception
$n(t) :$	Bruit à la réception
$S_i :$	$i^{me}$ symbole
$\tau :$	Délai
V :	Tension
W :	Largeur de bande

**LISTE DES TABLEAUX**

TABLE 1.1	Résumé des types de canaux sans-fil . . . . .	12
TABLE 1.2	Comparaison du BER et de l'efficacité spectrale pour différentes modulations avec une détection cohérente. . . . .	14
TABLE 3.1	Les séquences de Barker connues. . . . .	34
TABLE 3.2	Liste des paramètres du système utilisés lors de la simulation . . .	63

## LISTE DES ANNEXES

ANNEXE I	QUELQUES BASES THÉORIQUES . . . . .	125
I.1	Les récepteurs à maximum de vraisemblance [Proakis, J., 2000] . . . . .	125
I.1.1	Structure du récepteur . . . . .	125
I.1.2	Correspondance dans un espace vectoriel . . . . .	125
I.2	Performances d'une modulation . . . . .	128
I.2.1	Puissance de transmission et le SNR . . . . .	128
I.2.2	Taux d'erreurs dans un canal AWGN . . . . .	129
I.2.3	Efficacité spectrale . . . . .	132
I.3	Calcul de SNR à partir du MMSE . . . . .	132
I.4	Représentation de nombres binaires . . . . .	134
ANNEXE II	DETAILS D'IMPLÉMENTATION SUR SIMULINK . . . . .	136
II.1	Design du DSP . . . . .	136
II.2	Détails du FPGA du SMQUAD . . . . .	140
II.3	Détails du VHS-ADC en transmission . . . . .	144



## INTRODUCTION

La vision du sans-fil a beaucoup changé depuis le premier déploiement des systèmes de téléphonie cellulaire au début des années 1980. Une utilisation fractionnée du spectre de fréquences avait alors été considérée comme étant primordiale pour permettre une communication entre utilisateurs avec un minimum d'interférences. De nos jours, plusieurs organisations travaillent sur l'allocation et la réglementation du spectre des fréquences. Au Canada et aux États-Unis, Industrie Canada et la Federal Communications Commission en sont respectivement responsables. Au niveau global, l'International Telecommunications Union (ITU) est une agence des Nations Unies qui aide à coordonner l'utilisation du spectre des radiofréquences. Ces organismes ont dégagé certaines bandes de fréquence pouvant être utilisées sans license. Cette initiative qui avait pour but de permettre l'innovation et les implémentations à faible coûts a connu un grand succès. De nos jours, plusieurs systèmes de communications sans-fil opèrent à l'intérieur de ces bandes de fréquence. Ces technologies sont vastes, allant du réseau local sans-fil (WLAN) aux instruments Bluetooth et téléphones sans-fil.

L'industrie des semiconducteurs a aussi été en développement constant. En effet, l'implémentation d'algorithmes de haute complexité est de nos jours plus facilement atteignable avec les composantes matérielles (hardware) disponibles. Les composantes électroniques reprogrammables tels que les FPGA et les DSP utilisent une description logicielle (VHDL ou C++ embarqué) pour configurer aisément les algorithmes qu'ils réalisent. L'amélioration de ces outils de prototypage rapide a permis d'atteindre les objectifs majeurs suivants [Wakerly, J.F, 2001] :

- Limiter le temps de développement des produits. La configuration des composantes se fait à partir de la logique du design et ne requiert pas nécessairement des considérations au niveau de l'opération des transistors ou autres composantes analogiques.
- Diminuer les coûts associés à l'implémentation des solutions. Comme le comporte-

ment des circuits digitaux sont décrits par leur logique d'opération, la reproduction des circuits peut se faire de façon répétitive pour fournir des résultats similaires.

- Fournir des mesures préliminaires à l'étape de la faisabilité, limitant ainsi le risque associé au développement des produits complexes.
- Augmenter la vitesse de traitement des données. Plusieurs algorithmes nécessitent une grande vitesse de traitement des données offerte par les circuits intégrés pour atteindre de bonnes performances.

Cependant, avec cette croissance d'utilisation d'équipements sans-fil dans les bandes de fréquences sans license, une des craintes soulevées est le niveau des interférences générées et subies par les transmissions radio. Ces liens fonctionnent normalement à une puissance de transmission relativement basse et sont conçus pour fonctionner à courte portée. Cependant, dans une même zone plusieurs équipements sans-fil peuvent être en utilisation dans la même bande. Par exemple, l'utilisation des routeurs sans-fil pour l'accès à un réseau local et à Internet devient de plus en plus courant. Dans un même bâtiment résidentiel, plusieurs d'entre eux peuvent souvent être détectés. Comme l'interférence ainsi produite peut être non-négligeable, une dégradation du signal de réception pourrait s'ensuivre, ce qui peut détériorer la qualité de la transmission et de la performance du système de communications.

## **Objectifs**

Pour caractériser le type de dégradation selon différents environnements de propagation, des mesures de performance d'un lien sans-fil doivent être effectuées. Dans le cadre de ce projet, nous faisons le développement de la partie bande de base d'un système de communications sans-fil sur des plateformes de prototypage rapide de la compagnie Lyrtech Inc. Le projet est une étape préliminaire à l'élaboration d'un système de communications sans-fil avancé. Les objectifs que nous nous sommes fixés ici sont les suivants :

- Élaborer un modèle de communications sans-fil en bande de base de complexité moyenne.

- Vérifier la fonctionnalité du modèle à l'aide d'un outil de simulation et comparer sa performance par rapport aux résultats théoriques.
- Comprendre et documenter l'outil de développement fourni par Lyrtech Inc.
- Passer au premier stade de prototypage du système de communications sans-fil et en évaluer la performance.

Cette plateforme a été récemment acquise par le département de génie électrique de l'École Polytechnique de Montréal. Plusieurs autres universités à travers le monde ont développé leur prototype d'essai. Dans ce contexte, le projet marque donc le début de la conception du premier prototype d'un système de communications sans-fil pour le groupe de télécommunications du département. Il s'agit principalement de comprendre la programmation des composantes de la plateforme et l'utilisation des outils fournis dans le contexte de la communication sans-fil.

### **Organisation du mémoire**

Ce document décrit les étapes que nous avons suivies dans le cadre de ce projet. L'organisation de ce mémoire est comme suit. Au chapitre 1, nous faisons un rappel de certains concepts de base utilisés dans le cadre de ce projet. Ceci devrait permettre au lecteur d'avoir un bref aperçu des défis liés à la conception des systèmes de communications sans-fil.

Au chapitre 2, nous faisons une brève revue des techniques de prototypage de systèmes de communications utilisées et des systèmes réalisés à travers le monde scientifique.

Au chapitre 3, nous présentons le modèle du système et les différents algorithmes utilisés ainsi que les résultats obtenus par simulation. Nous décrivons les différents blocs élaborés et intégrés tant au transmetteur qu'au récepteur. Cette étape permettra de valider le système conçu ainsi que de prévoir les limites en terme de performance à laquelle s'attendre.

L'outil de prototypage rapide proposé par Lyrtech Inc est introduit au chapitre 4. Nous y décrivons les principales composantes matérielles et logicielles qui sont utilisées dans le cadre de ce projet.

Les parties implémentées sur la plateforme de prototypage sont décrites au chapitre 5. L'implémentation du transmetteur est détaillée. En ce qui a trait au récepteur, nous fournissons les démarches et concepts utilisés dans le design. Le filtre d'égalisation LMS ainsi que les algorithmes de synchronisation de phase et de recouvrement d'horloge sont implémentés. La méthodologie pour la collecte des données est décrite et l'analyse des mesures obtenues est aussi présentée dans ce chapitre. Ces résultats permettent la caractérisation du système et de valider sa fonctionnalité au niveau matériel.

Nous fournissons finalement des idées d'améliorations et des propositions pour des développements potentiels au chapitre 6.

## **CHAPITRE 1**

### **CONCEPTS DE BASE**

Les systèmes de communications point à point sont tous constitués de trois composantes : le transmetteur, le canal de transmission et le récepteur. Il existe plusieurs critères de performance pour un lien de communications. Pour les systèmes de communications numériques, l'objectif visé est de minimiser le taux d'erreurs binaires et de maximiser le taux de transmission sujet à la bande passante du canal et à la puissance de transmission. Dans ce chapitre, nous présentons les concepts de base nécessaires concernant les éléments affectant la performance d'un système de communications sans-fil.

Afin d'assurer une bonne fonctionnalité d'un système, il est utile de préalablement le simuler en intégrant les différents algorithmes utilisés. Ceci permettra d'avoir une bonne perspective sur les performances réalisables du système. Dans ce chapitre, nous mentionnons donc aussi les modèles utilisés pour caractériser chacun des éléments importants.

#### **1.1 Le canal sans-fil**

Un canal de communications peut être défini comme la liaison entre le transmetteur et le récepteur. De manière générale, il est perçu comme la source de plusieurs types de distortion du signal transmis. Ces distortions peuvent être caractérisées par les paramètres décrits dans les prochaines sections. De plus, un même canal peut être vu sous différentes perspectives selon le type de communication effectué, ce qui influencera le modèle utilisé pour évaluer la performance simulée du système conçu.

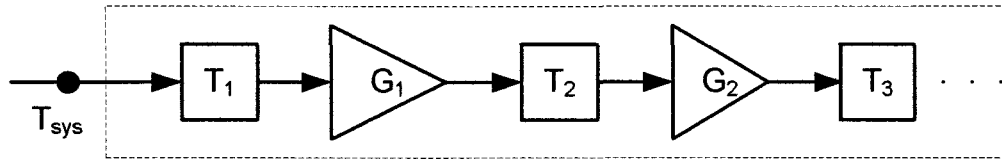


FIGURE 1.1 Température du bruit pour un système en cascade

### 1.1.1 Bruit additif

Le bruit présent au récepteur est un des principaux facteurs qui influencent la performance d'un système de communications. De façon générale, le bruit additif est considéré comme étant la somme de plusieurs sources de distortion. Le bruit thermique et le bruit de quantification en sont deux exemples classiques.

Le bruit thermique est généré par l'excitation des électrons provenant des composantes électroniques. La mécanique quantique suggère que la densité spectrale du bruit thermique est donnée par l'équation 1.1 [Alexander S. B, 1997].

$$S(f) = \frac{hf}{e^{\left(\frac{hf}{kT}\right)} - 1} \frac{\text{watts}}{\text{Hz}} \quad (1.1)$$

où  $h$  est la constante de Plank,  $f$  est la fréquence (Hz),  $k$  est la constante de Boltzmann et  $T$  est la température (Kelvin). Comme les systèmes de communications opèrent à des fréquences de l'ordre de quelques Gigahertz au plus et que la largeur de bande est relativement petite, nous pouvons, en utilisant la règle de l'Hopital, poser l'hypothèse que le bruit thermique est blanc, c'est-à-dire que la densité spectrale de puissance est uniforme [Alexander S. B, 1997].

$$S(f) \approx kT = N_0 \quad (1.2)$$

Un des paramètres utile pour le caractériser est la température effective de bruit. Pour les systèmes en cascade (figure 1.1, la température  $T_{sys}$  globale du bruit thermique peut être évaluée à partir des températures  $T_i$  individuelles de chacune des composantes ainsi que de leur gain  $G_i$ .

En appliquant l'équation 1.3, il sera donc possible d'avoir une bonne estimation du niveau de bruit au récepteur [Rappaport, T.S., 2002].

$$T_{sys} = T_1 + \frac{T_2}{G_1} + \frac{T_3}{G_1 G_2} + \dots \quad (1.3)$$

Le bruit de quantification est une autre source de distortion du signal qui provient de l'arrondissement de la valeur ou de la troncation des bits de données lors de la numérisation des calculs en points fixes. Le convertisseur analogique-numérique (ADC) est un des blocs où s'effectue l'arrondissement numérique des données. Pour un signal avec une tension crête de  $V_p$  quantifié sur  $b$  bits, la variance du bruit de quantification qui s'ajoute au système est donnée selon l'équation 1.4. Nous pouvons considérer que l'erreur de quantification suit une distribution uniforme [Sklar, B., 2001].

$$\sigma^2 = \frac{1}{3} \cdot \left( \frac{V_p}{2^b} \right)^2 \quad (1.4)$$

### **Modélisation**

Puisque le bruit additif provient de plusieurs sources, il est raisonnable d'utiliser le théorème Central Limite pour affirmer que la fonction de densité de probabilité du bruit suivra une distribution Gaussienne de moyenne nulle. Sa variance  $N_0$  équivaudra à la puissance du bruit. Par ailleurs, pour modéliser le bruit en bande de base, nous utilisons une variable aléatoire Gaussienne complexe circulaire,  $N$ . Ceci implique que la matrice de covariance  $K = N_0 I$  est nécessaire et suffisante pour complètement décrire les statistiques de  $N$  [Tse D. & Viswanath, P., 2005].

$$N \sim \mathcal{CN}(0, K) \quad (1.5)$$

Donc, selon la définition d'une statistique Gaussienne complexe circulaire, nous pouvons affirmer que  $N = N_x + jN_y$  avec  $N_x$  et  $N_y$  indépendants et définis par l'équation 1.6.

$$N_x \text{ et } N_y \sim \mathcal{N}(0, \frac{N_0}{2}) \quad (1.6)$$

La variable aléatoire  $X \sim \mathcal{N}(\mu, \sigma^2)$  est définie par la fonction de densité de probabilité fournie par l'équation 1.7.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad x \in \mathbb{R} \quad (1.7)$$

Par ailleurs, la densité spectrale du bruit varie selon le type de système conçu. Pour des systèmes de communications à bande étroite, il est raisonnable de considérer que la densité spectrale est constante puisque la largeur de bande est trop petite pour que les fluctuations dans le spectre puissent être observables. En d'autres mots, un modèle de bruit additif blanc Gaussien (AWGN) est convenable.

### 1.1.2 Atténuation

La propagation dans l'espace libre est utile pour estimer le niveau de puissance du signal reçu au récepteur en absence d'obstacles. Dans le cas général, nous pouvons affirmer que

$$P_r \propto d^{-n} \quad (1.8)$$

avec  $P_r$  la puissance du signal au récepteur,  $d$  la distance, et  $n$  un facteur indiquant le taux de décroissance de la puissance [Rappaport, T.S., 2002]. Pour une source isotropique se trouvant dans l'espace libre,  $n$  est égal à 2. En effet, la puissance est distribuée à une distance  $d$  sur une sphère de surface  $4\pi d^2$ , ce qui implique qu'avec un récepteur de surface



$A$ , la puissance capturée est de

$$P_r = \frac{P_t A}{4\pi d^2} \quad (1.9)$$

où  $P_t$  est la puissance transmise.

Dans d'autres situations,  $n$  peut varier de 1 à 6 selon l'environnement de propagation. Par exemple, des mesures effectuées dans [Saleh, A. & Valenzuela, R., 1987] mentionnent la valeur  $n = 3.9$  pour un bâtiment commercial avec le transmetteur et le récepteur se situant sur le même étage.

### **Modélisation**

L'atténuation peut être modélisée par un facteur de gain ou être incorporée dans le niveau de bruit relatif à la puissance du signal reçu. Son niveau dépend alors du type d'application projeté. Comme nous tentons de valider la performance du système, la pire atténuation possible sera utilisée. Elle sera l'atténuation provenant d'une source isotrope avec un récepteur se trouvant à la distance maximale désirable pour l'application.

#### **1.1.3 Multi-trajets**

Dans des systèmes de communications sans-fil, le signal transmis peut parcourir plusieurs trajectoires distinctes pour atteindre le récepteur tel qu'illustré à la figure 1.2. Plusieurs sources de multi-trajets sont décrites dans la littérature [Hashemi, H., 1993, Rappaport, T.S., 2002].

Lorsque le signal est réfléchi, l'énergie transmise est reliée au coefficient de réflexion de Fresnel,  $\Gamma$ . Celui-ci est fonction de plusieurs paramètres, notamment la polarization de l'onde, l'angle d'incidence, la fréquence de l'onde, et les propriétés du matériel.

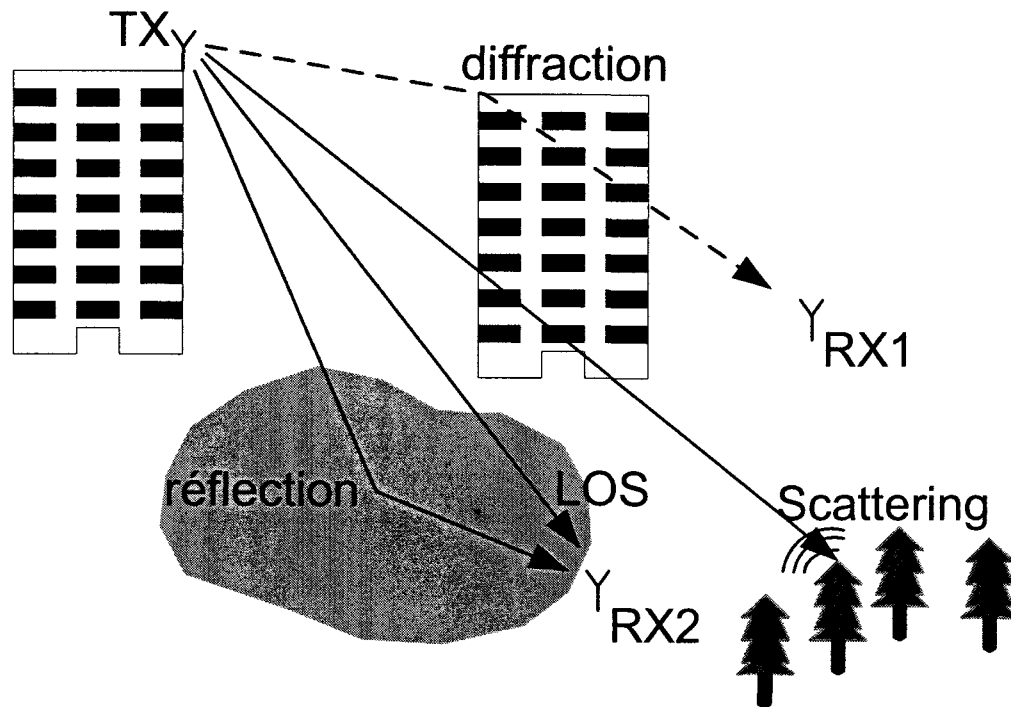


FIGURE 1.2 Les causes de multi-trajectoires

La diffraction permet à l'onde de se propager à travers des parcours obstrués. Même si la puissance du signal peut chuter de façon significative, elle peut être suffisante pour être utile au récepteur. Les études de l'effet de la diffraction peuvent être réalisées à travers différents modèles, notamment les zones de Fresnels et le *Knife-edge Diffraction model*.

Finalement, lorsque le signal est réfléchi sur une surface rugueuse, elle est dispersée dans toutes les directions, incluant vers le récepteur. Même si la puissance ainsi transmise est une fraction du signal incident sur la surface, elle est une cause non-négligeable de multi-trajets. Ce phénomène est souvent référé dans la littérature par la terminologie *scattering*.

Les trajets multiples sont la cause d'interférences constructives ou destructives. Ceci mène au phénomène d'évanouissement dans les systèmes de communications sans-fil.

## Modélisation

La variation temporelle et aléatoire de l'environnement de propagation rend l'analyse de l'effet des multi-trajets complexe. En effet, le signal de réception est la somme des différentes composantes du signal atténuées, déphasées et retardées. Une des méthodes proposées pour modéliser ce phénomène est de considérer le canal comme un filtre linéaire [Hashemi, H., 1993]. Grâce à celui-ci, pour un signal de transmission en bande de base  $s(t)$ , le signal reçu sera donné par l'équation 1.10.

$$r(t) = \Re \left[ \sum_{k=0}^{N-1} a_k s(t - \tau_k) e^{j(\omega_0 t + \theta_k)} \right] \quad (1.10)$$

où  $a_k$ ,  $\tau_k$  et  $\theta_k$  sont l'atténuation le délai subi et la phase respectivement de la  $k^{ime}$  composante,  $\omega_0$  est la fréquence angulaire de la porteuse et  $N$  est le nombre de multi-trajets. Une analyse de la dynamique du système [Tse D. & Viswanath, P., 2005] permet de définir les métriques suivantes :

### Temps de cohérence, $T_c$

Il définit la durée pour laquelle les paramètres du canal ne changent pas significativement.

### Étalement de Doppler, $D_s$

L'étalement de Doppler indique le taux auquel le patron d'interférence change. Il est inversement proportionnel au temps de cohérence.

### Étalement de délai, $T_D$

Ce paramètre est défini comme étant la différence maximale entre les temps de propagation du signal. Comme il n'est pas toujours pratique de le mesurer, d'autres paramètres, tel que l'étalement de délai rms (rms delay spread),  $\tau_{rms}$ , peuvent être utilisés.

Type de Canal	Caractéristique
Non-sélectif	$W \ll W_c$
Sélectif en fréquence	$W \gg W_c$
Évanouissement lent	$T_{trame} \ll T_D$
Évanouissement rapide	$T_{trame} \gg T_D$

TABLE 1.1 Résumé des types de canaux sans-fil

### Bande de Cohérence, $W_c$

Elle définit la largeur de la bande de fréquence pour laquelle le gain du canal reste quasiment constant. Elle est reliée à l'étalement de délai par la formule 1.11 :

$$W_c \propto \frac{1}{2T_d} \quad (1.11)$$

Le tableau 1.1 décrit la classification des canaux sans-fil en fonction de ces quatre paramètres.  $T_{trame}$  est la durée d'une trame de données alors que  $W$  est la largeur de bande occupée par le signal transmis. Comme nous le constatons au tableau 1.1, les caractéristiques du canal dépendent de la relation entre les propriétés du signal transmis par rapport aux paramètres du canal. Par exemple, le même canal peut être sélectif en fréquences pour des techniques de transmission utilisant de larges bandes de fréquences mais être non-sélectif pour les transmissions à bande étroite. À noter qu'un canal non-sélectif en fréquence peut être modélisé par un canal avec une seule composante de multi-trajet (i.e  $N = 1$ ) [Tse D. & Viswanath, P., 2005].

#### **1.1.4 Interférences**

Les deux principales sources d'interférences sont l'interférence entre symboles (ISI) et l'interférences provenant des liens externes. L'ISI provient des effets de filtrage existant dans les systèmes de communications. Par exemple, les canaux sans-fil sont souvent caractérisés par une fonction de transfert, c'est-à-dire qu'ils agissent comme des filtres. Ny-

quist a étudié le problème de la communication sans ISI dans les canaux à bande limitée (Nous en discuterons davantage à la section 3.3).

Par ailleurs, il est possible qu'un système de communications avoisinant opère dans la même bande de fréquence (co-channel interference) ou proche (adjacent channel interference). Ces types d'interférences peuvent être limités à travers une bonne planification du réseau sans-fil [Sklar, B., 2001]. D'un point de vue pratique, il est raisonnable dans la plupart des cas de considérer les interférences externes comme étant des sources de bruit additif.

## **1.2 La modulation**

La modulation consiste à effectuer l'encodage des données binaires selon un signal ou une combinaison d'un ensemble de signaux de base. Cet ensemble de signaux de base est habituellement orthonormal. Ainsi les signaux de transmission possibles pourront être modélisés comme des vecteurs dans un espace vectoriel où chaque axe correspondra à une des fonctions orthonormales. Chaque séquence binaire possible est alors assignée à un point de la constellation.

Le démodulateur au récepteur doit estimer la séquence transmise. Une des techniques utilisées est de décider selon le point de la constellation le plus proche du vecteur reçu. En d'autres termes, ce récepteur, défini comme le récepteur à maximum de vraisemblance (Maximum Likelihood Receiver), décide de la combinaison des signaux transmis en faisant une comparaison avec ceux auxquels il s'attend. Le récepteur à maximum de vraisemblance est davantage détaillé dans l'annexe I.1.

Le choix de la technique de modulation utilisée est principalement basé sur les considérations suivantes [Goldsmith A., 2005] :

- le taux de transfert ;

Modulation	BER	Efficacité Spectrale (bits/s/Hz)
BPSK	$Q(\sqrt{2SNR})$	1
QPSK	$Q(\sqrt{SNR})$	2
M-PSK	$\frac{2}{\log_2 M} Q(\sqrt{2SNR} \sin(\pi/M))$	$\log_2(M)$
M-QAM	$\frac{4(\sqrt{M}-1)}{\sqrt{M} \log_2 M} Q(\sqrt{\frac{3SNR}{M-1}})$	$\log_2(M)$

TABLE 1.2 Comparaison du BER et de l'efficacité spectrale pour différentes modulations avec une détection cohérente.

- l'efficacité spectrale ;
- l'efficacité de puissance ;
- le taux d'erreurs ;
- le coût d'implémentation.

Par exemple, tel qu'indiqué dans le tableau 1.2, le taux d'erreurs binaires obtenu par une modulation BPSK est inférieur à celui d'une modulation QPSK. Il y a une perte de 3dB en termes de rapport signal à bruit ( $SNR = \frac{E_s}{N_0}$ ) requis pour atteindre le même taux d'erreurs. Cependant, l'efficacité spectrale est meilleure pour une modulation QPSK. Le choix de la modulation nécessite donc clairement un compromis entre les divers facteurs mentionnés.

À noter que dans le calcul du taux d'erreurs binaires (BER), nous avons assumé qu'un ordonnancement de Gray est utilisé. Dans un ordonnancement de Gray, la correspondance binaire de chaque symbole de la constellation est assignée de sorte à ce que chaque symbole se distingue de son voisin le plus proche par au plus 1 bit.

### 1.3 La conversion en bande passante

Avant la transmission des données à travers le canal sans-fil, une modulation vers des radio-fréquences est généralement effectuée. Une des justifications pour cette opération est que la dimension requise des antennes de transmission et de réception est proportionnelle à la longueur d'onde du signal. Par exemple, en assumant que la dimension de

l'antenne doit être de  $\lambda/4$ , pour transmettre une longueur d'onde de 30kHz, une antenne irréaliste de 2500m serait nécessaire. À 600MHz, elle sera de 12.5cm.

Il est cependant désirable d'effectuer l'analyse et le traitement des signaux en bande de base car le coût du matériel requis ainsi que la complexité de calculs sont ainsi considérablement réduits. La représentation en quadrature est alors utile. Elle établit une équivalence entre les signaux bande de base et bande-passante facilitant ainsi l'analyse et le traitement de données en bande de base.

### Représentation en quadrature

Soit un signal bande de base  $m(t) = x(t) + jy(t)$  de largeur de bande limitée à  $W$ . La modulation en quadrature peut être vue comme étant la multiplication de  $m(t)$  par  $e^{j2\pi f_0 t}$  suivi de la transmission de la partie réelle du signal. En écrivant  $m(t)$  comme  $x(t) + jy(t)$ , la relation mathématique suivante peut être établie.

$$\begin{aligned} s(t) &= \text{Re}\{m(t)e^{j2\pi f_0 t}\} \\ &= \text{Re}\{(x(t) + jy(t))(\cos 2\pi f_0 t + j \sin 2\pi f_0 t)\} \\ &= x(t)\cos 2\pi f_0 t - y(t)\sin 2\pi f_0 t \end{aligned} \quad (1.12)$$

Au récepteur, les multiplications de  $s(t)$  par  $\cos 2\pi f_0 t$  et  $-\sin 2\pi f_0 t$  suivi d'un filtre passe-bas génèrent respectivement la partie réelle et imaginaire de  $m(t)$ .

$$\begin{aligned} r_{\text{real}}(t) &= s(t) \cos 2\pi f_0 t|_{\text{passe-bas}} & r_{\text{imag}}(t) &= s(t) (-\sin 2\pi f_0 t)|_{\text{passe-bas}} \\ &= \frac{x(t)}{2}(1 + \cos 4\pi f_0 t) & &= \frac{y(t)}{2}(1 - \cos 4\pi f_0 t) \\ &\quad - \frac{y(t)}{2}\sin 4\pi f_0 t|_{\text{passe-bas}} & &- \frac{x(t)}{2}\sin 4\pi f_0 t|_{\text{passe-bas}} \\ &= \frac{x(t)}{2} & &= \frac{y(t)}{2} \end{aligned} \quad (1.13)$$

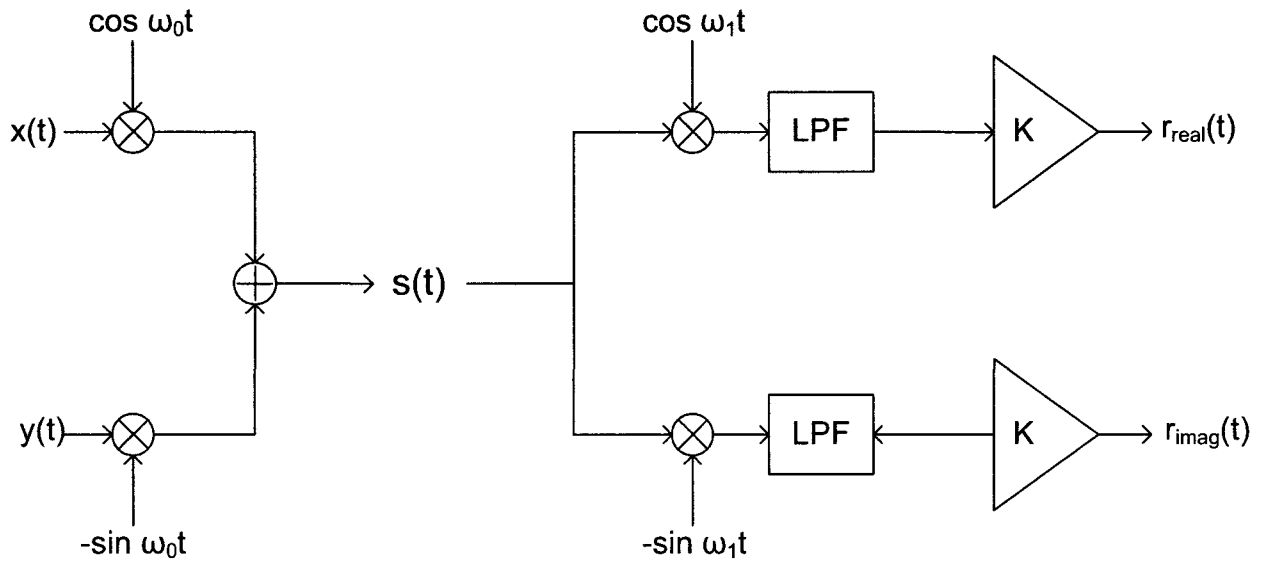


FIGURE 1.3 Conversion de bande de base à bande passante au transmetteur et de bande passante à bande de base au récepteur

La figure 1.3 illustre les étapes décrites ci-haut. À noter que même si le signal de réception  $r(t)$  a été atténué, il n'y a pas de perte d'information et donc la capacité n'est pas affectée. En effet, il est toujours possible d'amplifier le signal par un facteur de deux pour compenser ces atténuations.

#### 1.4 La période d'un symbole

Combiné avec le type de modulation utilisé, la période d'un symbole  $T_s$  détermine le taux de transfert des informations binaires. Pour une modulation QPSK, deux bits de données sont représentés par symbole. Donc, pour une période  $T_s$  secondes, le taux de transfert est de  $2/T_s$  bits/s. Par contre, pour une constellation M-QAM,  $\log_2 M$  bits de données sont représentés par symbole. Le taux de transfert est alors de  $(\log_2 M)/T_s$  bits/s.



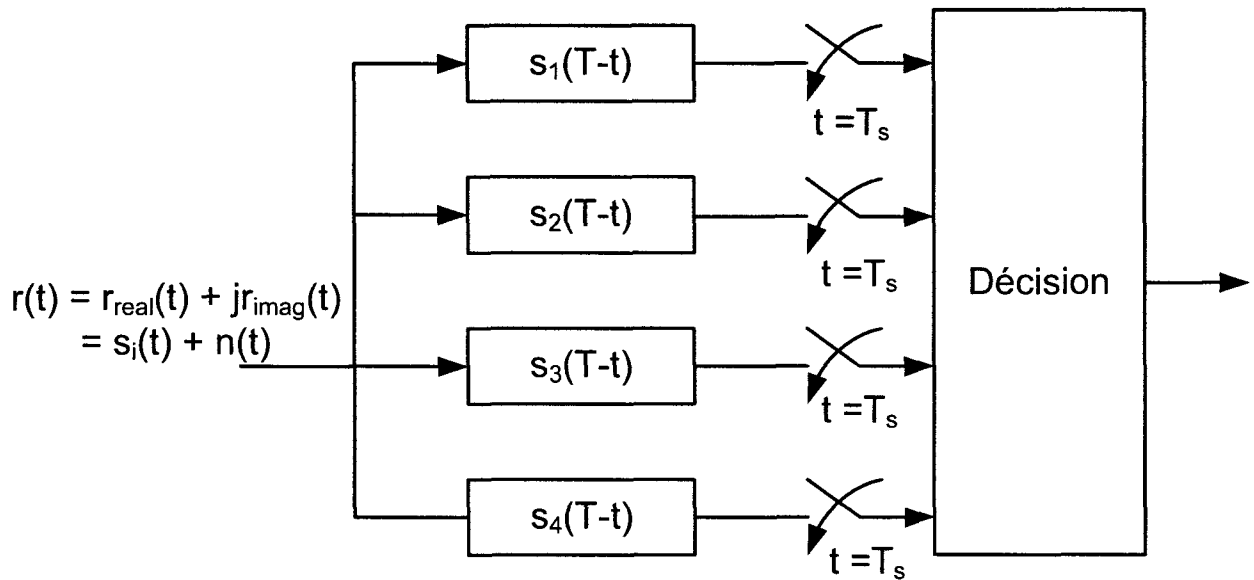


FIGURE 1.4 Structure d'un récepteur optimal

### 1.5 Détection optimale

Au transmetteur, le modulateur se base sur la combinaison des données binaires pour envoyer un des  $M$  signaux possibles. Au récepteur, le défi est de faire la meilleure estimation du signal transmis pour décider du message. Un récepteur qui permet de maximiser le rapport signal à bruit permettra de faire la meilleure décision. La probabilité d'erreurs sera alors minimisée. Ce récepteur optimal comprend un ensemble de filtres linéaires adaptés aux signaux transmis (voir figure 1.4). Comme chacun de ces filtres est équivalent à un corrélateur, le récepteur optimal est aussi un récepteur à maximum de vraisemblance tel que décrit à l'Annexe I.1 [Wozencraft, J.M & Jacobs, I.M., 1965].

À la figure 1.4, nous observons que la sortie du filtre doit être échantillonnée à une période  $T_s$  correspondant à la durée d'un symbole. Un des critères importants pour la détection optimale est de faire l'échantillonnage des filtres adaptées au bon instant. Comme nous l'observons à la figure 1.5, dans un système de communication numérique, le convertisseur analogique-numérique effectue la discrétisation du signal reçu. De façon générale, ce convertisseur échantillonne le signal de réception à un facteur  $L$  de la période  $T_s$ . Le fac-

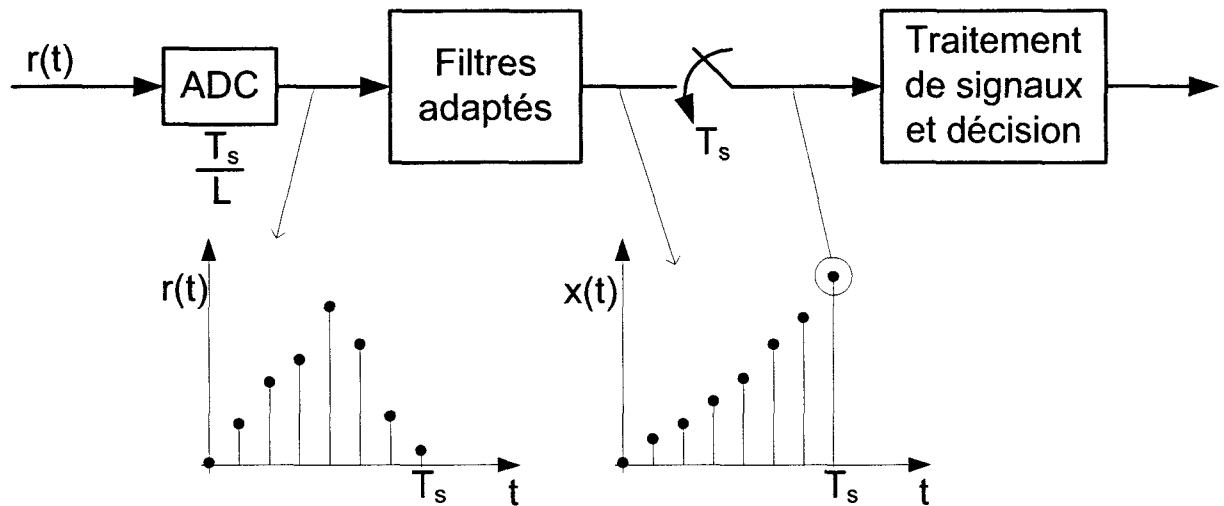


FIGURE 1.5 Discrétisation du filtre adapté

teur  $L$  est le taux de suréchantillonnage. Chaque suréchantillon à la sortie du convertisseur passe à travers le filtre adapté. Le bon échantillon de chaque groupe de  $L$  suréchantillons devra alors être choisi. Dans l'exemple de la figure 1.5, un signal BPSK est transmis et le signal reçu  $r(t)$  est suréchantillonné par le convertisseur analogique numérique. Un corrélateur effectue la corrélation des suréchantillons avec la fonction de base utilisée et à sa sortie, l'instant optimal d'échantillonnage est lorsque  $t = T_s$ . Il est important de sélectionner le bon échantillon à la sortie des filtres adaptés pour maximiser le rapport signal à bruit et donc minimiser le taux d'erreurs.

## 1.6 L'horloge

Dans les systèmes de communications numériques, les données sont transmises par des symboles de durée  $T_s$  secondes. Cependant, le délai subi par le signal à travers le canal n'est pas nécessairement un multiple de  $T_s$ . Ainsi, l'échantillonnage du signal reçu ne sera pas effectué de façon synchrone. La figure 1.6 illustre ce phénomène. Plusieurs blocs au récepteur requièrent l'échantillonnage à des instants  $kT$ , pour des valeurs de  $k$  entières, du signal de réception. La démodulation optimale en est un exemple. De plus, si

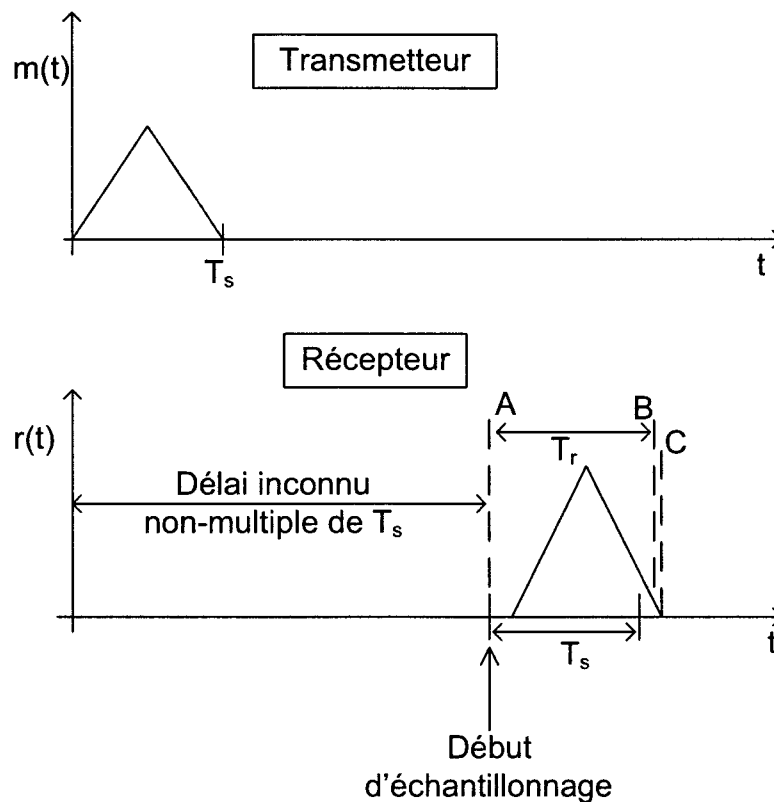


FIGURE 1.6 Effet du délai de canal sur le moment d'échantillonnage

l'instant d'échantillonnage du signal de réception est inexact, il y aura de l'interférence inter-symboles pour un canal à bande limitée [Sklar, B., 2001]. À priori, nous pouvons affirmer que le récepteur n'a pas de notion du moment exact auquel l'échantillonnage être effectuée. Dans le cas présenté à la figure 1.6, celui-ci se passe en avance (au point A) par rapport au bon instant (Point C).

De plus, la durée d'une période d'une symbole au récepteur  $T_r$  peut être légèrement différente de celle au transmetteur dû aux imperfections des oscillateurs. À la figure 1.6, le récepteur considère qu'un symbole dure pendant une période  $T_r$  qui est supérieure à la vrai durée  $T_s$  d'un symbole. À long terme, cette erreur peut s'accumuler et nécessite donc d'être corrigée.

## **Modélisation**

En insérant un nombre aléatoire de délais de période équivalent à  $T_s/L$  dans le modèle de canal (avec  $T_s$  la période d'un symbole et  $L$  le facteur de suréchantillonnage), nous pouvons considérer que le récepteur aura perdu la notion du temps d'arrivée exact des données. Ainsi, cela pourra servir de test pour vérifier la validité d'un algorithme de recouvrement d'horloge.

### **1.7 Oscillateurs**

La génération des fonctions cycliques tels que l'horloge électronique et les sinusoïdaux se fait souvent à partir d'un cristal. Cependant, comme ceci n'est pas parfait, il existe une déviation entre sa fréquence nominale et réelle. Ceci fait que les cristaux au transmetteur et au récepteur oscilleront à des fréquences légèrement différentes. En conséquence, il existera une différence entre la fréquence de la porteuse reçue et de la sinusoïdale de référence (voir figure 1.3). Le passage des radio-fréquences en basse fréquence comportera donc une erreur, ce qui devra être compensée par un algorithme de synchronisation de fréquence.

## **Modélisation**

Afin d'inclure cette disparité, nous avons considéré que la différence entre les fréquences équivaut à une sinusoïdale d'erreur de fréquence  $f_e$  modulant le signal dans le canal, la fréquence  $f_e$  étant égale à  $(f_{transmetteur} - f_{recepteur})$ , où  $f_{transmetteur}$  est la fréquence au transmetteur et  $f_{recepteur}$  est la fréquence au récepteur.

## **1.8 Système global**

La figure 1.7 permet de résumer le système de communications. Au transmetteur, les données binaires sont traitées pour former des paquets qui seront transmis. Le canal de transmission réel est constitué de plusieurs facteurs mais la simulation d'un système ne requiert qu'une modélisation partielle mais suffisante du système.

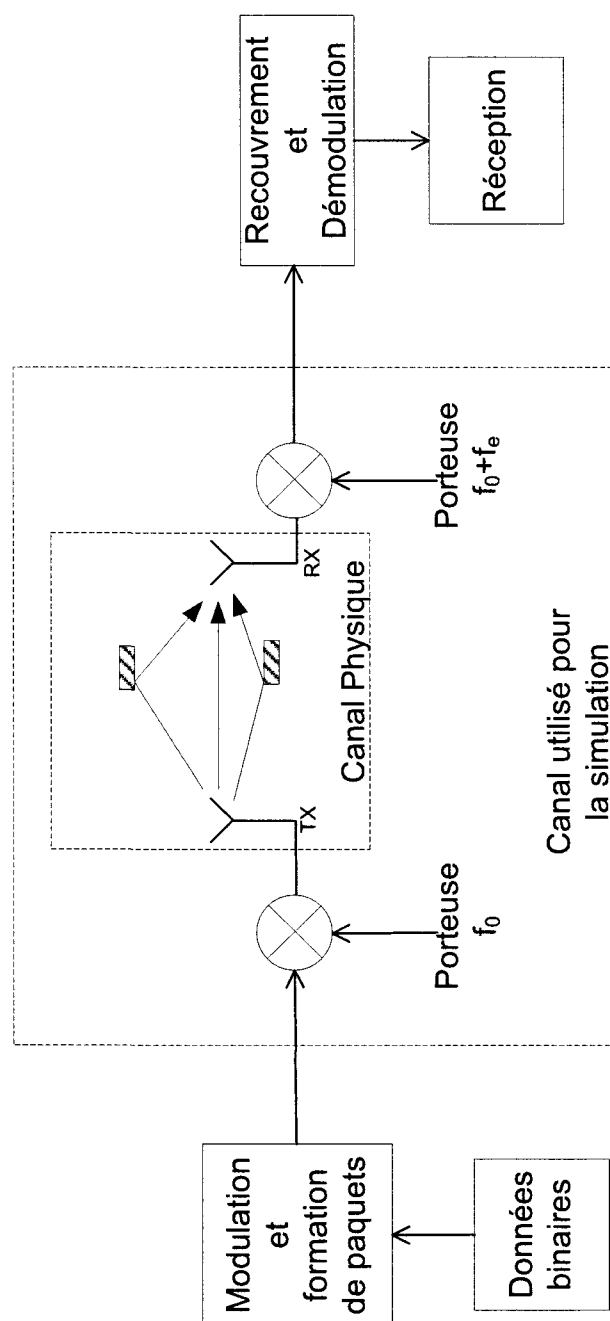


FIGURE 1.7 Une vue de haut niveau du système de communications

## CHAPITRE 2

### SURVOL DES TECHNIQUES DE PROTOTYPAGE DE SYSTÈMES DE COMMUNICATIONS

La conception des systèmes de communications sans-fil fait face à plusieurs défis. Au courant de ces dernières années plusieurs nouveaux concepts ont été élaborés. Par exemple, les études sur les systèmes à multiples antennes (MIMO) sont très répandues. Une bonne étude théorique fournit une base solide à partir de laquelle plusieurs idées peuvent être développées. La simulation des systèmes conçus fournit la possibilité de vérifier leurs fonctionnalités et les performances prévues en se basant sur des modèles avant de passer à l'implémentation matérielle. Dans ce chapitre, nous faisons un bref aperçu de quelques systèmes de prototypage qui ont été réalisés au courant de ces dernières années.

#### 2.1 Critères de développement

La réalisation matérielle d'un système de communications requiert un certain niveau d'ingénierie qui pouvait être ignorée lors des études théoriques. Ainsi, elle est assujettie à plusieurs contraintes qui influencent la faisabilité du produit. Quelques critères de décision sont les suivants [Raghu M.R. et al, 2004, Safadi M.S. & Ndzi D.L., 2006] :

- **Coût.** Ce coût peut être mesuré en argent, en temps de développement ou en termes de consommation de puissance.
- **Flexibilité.** Aux stades préliminaires de développement, une bonne configurabilité est importante pour différentes raisons. Par exemple, cela permettra de valider aisément certains algorithmes. De plus, un vaste choix de fonctionnalités est désirable.
- **Débit binaire.** Le débit qui peut être atteint est classé selon deux catégories : le transfert

temps-réel et non temps-réel. Un transfert en temps réel est beaucoup plus demandant en termes de besoin de traitement et de développement qu'un transfert non temps réel, mais il est plus représentatif des différentes applications de systèmes de communications existants.

## 2.2 La radio logicielle

Il est clair que les besoins en termes de communications sans-fil ont beaucoup évolué au courant de ces dernières années. De nos jours, les produits sans-fil sont utilisés pour une panoplie d'activités, notamment les communications voix et vidéo, les transferts de données et le contrôle à distance. La radio logicielle (Software Defined Radio, SDR) est définie comme tout système qui permet d'avoir une communication sans-fil et qui contient au moins une fonction de la couche physique définie par une composante logicielle [SDR Forum]. Elles sont attirantes pour les raisons suivantes [Nakajima, N. et al, 2001] :

- **Reconfigurabilité.**

Présentement, les logiciels utilisés sur les stations mobiles sont principalement utilisés dans la couche application (Jeux, etc.). Un des objectifs futurs est de permettre la configuration de certaines autres fonctions. Par exemple, à une station de base, l'allocation des horaires de transmission (time slots) peut être modifié sans avoir à faire des changements matériels.

- **Opération Multimode.**

Les systèmes sans-fil de troisième (3G) et de quatrième (4G) génération sont présentement en déploiement à travers la planète. Il est essentiel d'être en mesure de fournir du service aux utilisateurs qui font beaucoup de déplacements indépendamment de leur emplacement. Dans certains cas, l'utilisateur pourrait même se retrouver en mode roaming à un endroit où seulement le 2G est supporté. Donc, les terminaux multi-modes sont désirables.



– **Déverminage et mise à jour logicielle.**

Beaucoup de terminaux mobiles sont déployés de nos jours. Lorsque la correction d'un bug est trouvée, il peut être coûteux de passer à son application tant au niveau de la station de base qu'au niveau du mobile si des changements matériels sont nécessaires. Avec les SDR, le matériel d'une station de base peut être reprogrammable. Cela implique alors une diminution des coûts de réparation et du temps d'indisponibilité. Le problème survient surtout au niveau des stations mobiles où la quantité déployée peut être énorme. Un rappel de ces équipements pour effectuer le changement matériel peut être alors coûteux. Le remplacement à distance du logiciel (lorsque possible) devient ainsi une option intéressante.

Plusieurs approches ont été utilisées pour le développement des radios logicielles. Dans les sections suivantes, nous faisons un bref aperçu des méthodes proposées dans la littérature.

### **2.3 Traitement hors ligne**

Dans les phases initiales de développement de nouveaux systèmes, le traitement hors ligne des données permet de valider des résultats théoriques et de simulation pour des environnements réalistes. Ceci offre la possibilité de faire études de performance et ajustements nécessaires préalables à l'intégration matérielle [Raghu M.R. et al, 2004].

Le système de [Lang, S. et al., 2004] est une plateforme destinée à concevoir un système SISO-OFDM. Le système réalisé est une première phase de développement et permettait d'avoir un système non-temps réel. Dans une phase subséquente, des systèmes MIMO sont prévus.

Le système fonctionne avec une porteuse de 5,25 GHz. Au niveau du transmetteur, les données ont été transmises dans la mémoire vive et transmises en paquets vers des conver-

tisseurs analogiques-numériques puis la section RF. Au niveau du récepteur, les données sont gardées en mémoire pour être traitées hors-ligne.

Le traitement de données se fait hors ligne avec différents algorithmes. Un tel système permet d'activer/désactiver certains blocs et de réajuster certains paramètres avec un temps de développement minimal. Le coût de développement est relativement faible. Cependant, il peut être utilisé seulement pour des mesures de systèmes à bande étroite. De plus, le traitement de données n'est pas effectué en temps réel.

## 2.4 DSP

Dans plusieurs situations, le développement d'un système dans un environnement temps réel est requis pour l'étude de performances. Une plateforme avec un DSP (Digital Signal Processor) permet d'atteindre cet objectif.

Un des systèmes réalisés sur DSP est présenté par [Stuber G.L. et al, 2004]. La configuration utilisée est présentée à la figure 2.1. L'utilisateur communique à chacun des deux cartes de développement à travers une interface VME à PCI. L'architecture avait été proposée par [Mitola, J., 1995]. Un système MIMO-OFDM  $2 \times 2$  est réalisé avec un codage d'Alamouti. Au récepteur, les différentes opérations à réaliser (synchronisation temporelle et fréquentielle, estimation et compensation de canal & démodulation) sont distribuées aux quatre processeurs présents sur la plateforme.

L'auteur mentionne que la prochaine phase de développement est l'utilisation des FPGA disponible pour effectuer du FEC (Forward Error Correction).

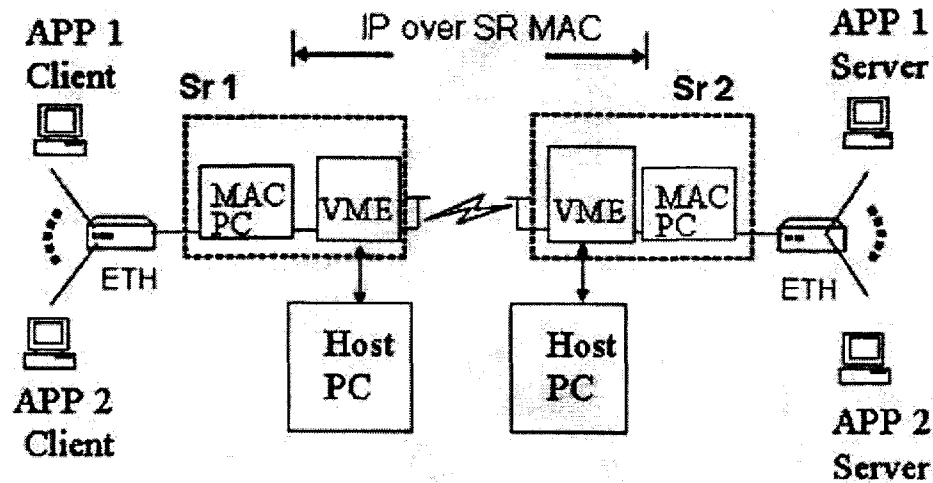


FIGURE 2.1 Plateforme pour un SDR sur DSP [Stuber G.L. et al, 2004]

## 2.5 ASIC

Un circuit ASIC (Application Specific Integrated Circuit) est un circuit intégré spécialisé pour effectuer certaines fonctions. Le coût de développement et la complexité associés avec ce type de produit sont élevés [Safadi M.S. & Ndzi D.L., 2006]. Les meilleures performances des puces ASIC offrent la possibilité d'effectuer des opérations temps réel pour des systèmes très demandant [Raghu M.R. et al, 2004]. De plus, le coût de production par échantillon est réduit par la suite.

L'article [Frigon J.-F. & Daneshrad, B., 2002] présente un système qui utilise la technologie ASIC. Au transmetteur, le signal est modulé à une fréquence IF de 43,75 MHz. La fréquence de la porteuse utilisée est de 2,44GHz. Au récepteur, une série de quatre antennes de réception espacées de  $\frac{1}{2}\lambda$  est utilisée. Le récepteur est configuré de sorte à avoir une série d'antennes intelligente. Toutes les fonctions du récepteur (traitement de signaux spatio-temporel, filtrage et synchronisation) ont été implémentées dans un seul ASIC. Les auteurs ont pris un total de 59 262 mesures pour un environnement de propagation intérieur et une communication point à point. Plusieurs résultats ont été déduits à partir de ces mesures. Par exemple, avec une puissance de transmission de 10 mW, le

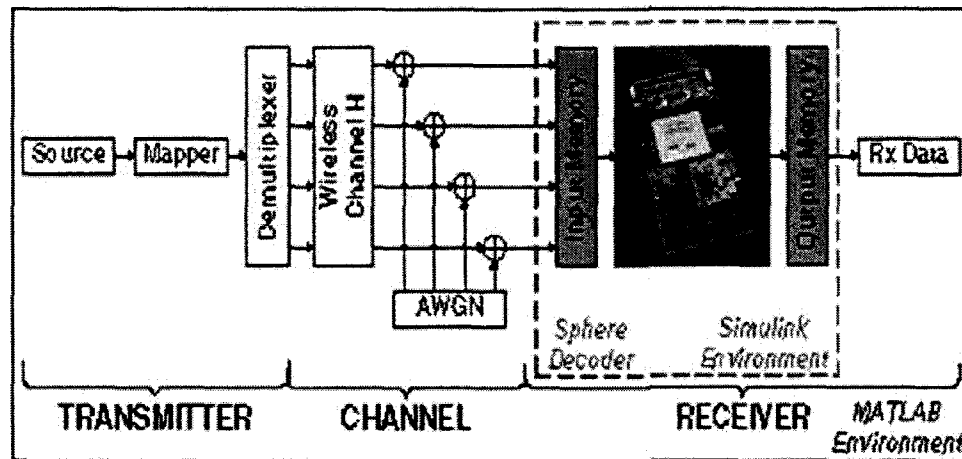


FIGURE 2.2 Modèle d'un système FPGA [Barbero, L.G. & Thompson, J.S., 2005]

système était en mesure de fournir un taux de transfert de 30 Mbps avec un taux d'erreurs binaires de  $10^{-3}$ , une probabilité de blocage de 5% sur un rayon de couverture de 20 m.

## 2.6 FPGA

L'utilisation d'un FPGA (Field Programmable Gate Array) utilise les avantages des systèmes ASIC tout en limitant le coût de développement. De plus, le FPGA offre une bonne réadaptabilité car il peut être reprogrammé pour faire des modifications au design.

Un tel système est présenté dans [Barbero, L.G. & Thompson, J.S., 2005]. L'objectif est d'implémenter et de faire l'analyse de plusieurs algorithmes pour des systèmes à multiples antennes (MIMO). Un schéma bloc du système utilisé est illustré à la figure 2.2.

L'outil Xilinx DSP System Generator est utilisé pour faire l'implémentation à partir d'un environnement Simulink de Matlab. Après avoir identifié la partie utile du modèle Simulink, cette dernière est isolée et compilée pour générer un bitstream. Le modèle FPGA est alors simulé comme un système embarqué sous un environnement Simulink. Cette technique est surnommée le Hardware in the Loop. Elle permet de simuler un morceau du

système et de changer son entrée selon les besoins à travers un interface usager (GUI) de Simulink. Les résultats correspondent aux données obtenues suite à un traitement en temps réel.

Une des reproches des FPGAs est l'utilisation inefficace de la logique disponible, ce qui implique une consommation excessive de puissance [Safadi M.S. & Ndzi D.L., 2006].

## 2.7 Prototypes hybrides

Les prochaines générations de SDR utilisent des architectures hybrides en combinant les différentes technologies disponibles. Comme mentionné à la section 2.4, une plateforme utilisant quatre DSP et deux FPGA est envisagée pour effectuer du MIMO-OFDM avec FEC.

Une architecture utilisant des FPGA et DSP est aussi mentionnée sous l'acronyme BEE (Berkeley Emulation Engine) par [Takai, M. et al, 2005]. Un environnement Simulink est utilisé pour développer les modèles. Ils peuvent ensuite être exécutés et la réalisation de la radio se fait par du prototypage rapide.

Il existe d'autres bancs d'essai en développement ou sur lesquels des systèmes ont été implémentés. Un banc d'essai utilisant une combinaison d'ASICs avec plusieurs FPGA de haute vitesse est en développement selon ce qui est mentionné par [Takai, M. et al, 2005]. Le but est de créer un système pouvant supporter des configurations MIMO  $4 \times 4$  pour un signal ayant une bande passante de 25 MHz et du traitement en temps réel.

Nous avons utilisé une plateforme de prototypage hybride. La plateforme est constituée de deux FPGA et de quatre DSP. Elle est davantage décrite au chapitre 4.

## CHAPITRE 3

### LES ÉLÉMENTS DU MODÈLE DE SIMULATION

Dans ce chapitre, nous décrivons les algorithmes utilisés au niveau du transmetteur et du récepteur de notre système de communications sans-fil. Le système est conçu pour fonctionner à bande étroite pour des liens de communications intérieur (indoor). La distance entre le transmetteur et le récepteur est considérée fixe et l'environnement de propagation est assumée quasi-statique. Les hypothèses posées sont les suivantes :

- Comme la transmission se fait à bande étroite, le canal est modélisé non-sélectif en fréquence. Le modèle du canal est donc constitué d'un seul tap.
- Comme la vitesse relative entre le transmetteur et le récepteur est négligeable et que l'environnement de propagation ne change pas beaucoup, nous en déduisons que le canal est à évanouissement lent.
- En assumant une distance maximale de  $100m$  entre le transmetteur et le récepteur, nous avons estimé une atténuation maximale entre le transmetteur et le récepteur de l'ordre de  $-60dB$ .
- En se basant sur quelques feuilles de spécifications provenant de plusieurs manufacturiers, nous avons assumé que la pire stabilité du crystal est de 5ppm [Crystals1, Crystals2, Crystals3].

Dans les sections suivantes, nous détaillerons les blocs utilisés pour réaliser le système de transmission. Pour chaque bloc, il existe plusieurs algorithmes possibles. Nous nous sommes limités à avoir un modèle de complexité moyenne et de performance acceptable.

La figure 3 illustre le schéma bloc du transmetteur bande de base. Les données binaires de la source sont modulées et des paquets constitués d'une certaine entête et d'une quantité fixe de données sont formés. Par la suite, chaque paquet est suréchantillonné par un facteur  $L$ . Les signaux sont filtrés avant la transmission vers le convertisseur analogique-

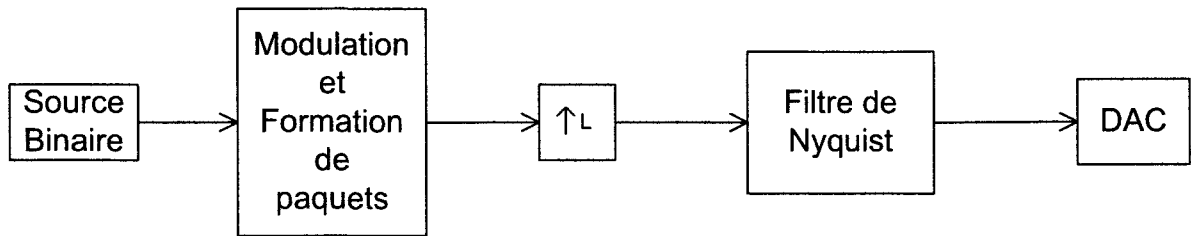


FIGURE 3.1 Schéma bloc du transmetteur

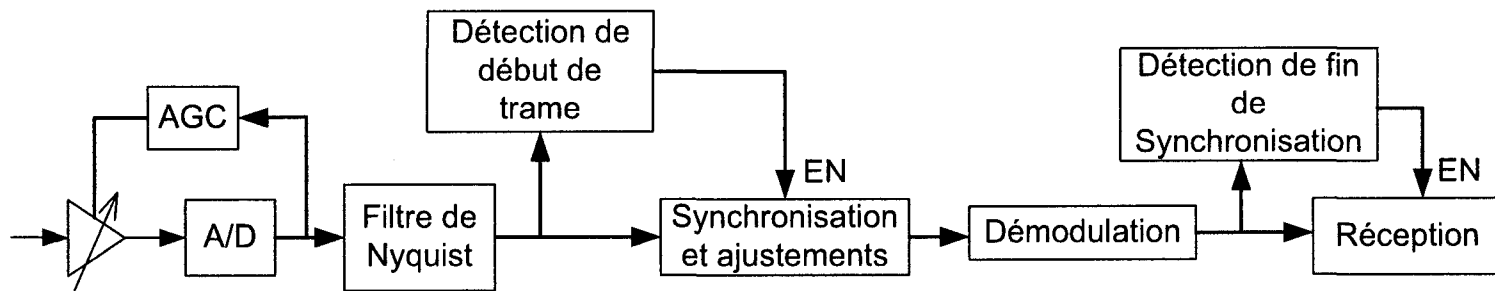
numérique et éventuellement vers la section haute fréquence du transmetteur.

Le récepteur requiert un design relativement plus complexe que le transmetteur. Avant de faire la démodulation du signal reçu, le récepteur doit effectuer un apprentissage du canal en utilisant un circuit d'ajustement de gains et des boucles de synchronisation. En effet, il faut que le récepteur soit en mesure d'estimer les effets du canal afin de corriger le signal en conséquence et d'effectuer la détection cohérente. À la figure 3.2(a), nous présentons la structure générale du récepteur en schémas blocs. Un circuit AGC contrôle l'amplitude du signal de réception. Par la suite, des blocs de synchronisation dont l'agencement est détaillé à la figure 3.2(b) effectuent d'autres corrections sur le signal. Ce circuit est activé lorsque le début d'une trame est détecté et y reste pendant la durée de la trame. Éventuellement, la démodulation s'ensuit. L'enregistrement des données reçues débute à partir de la fin de la synchronisation.

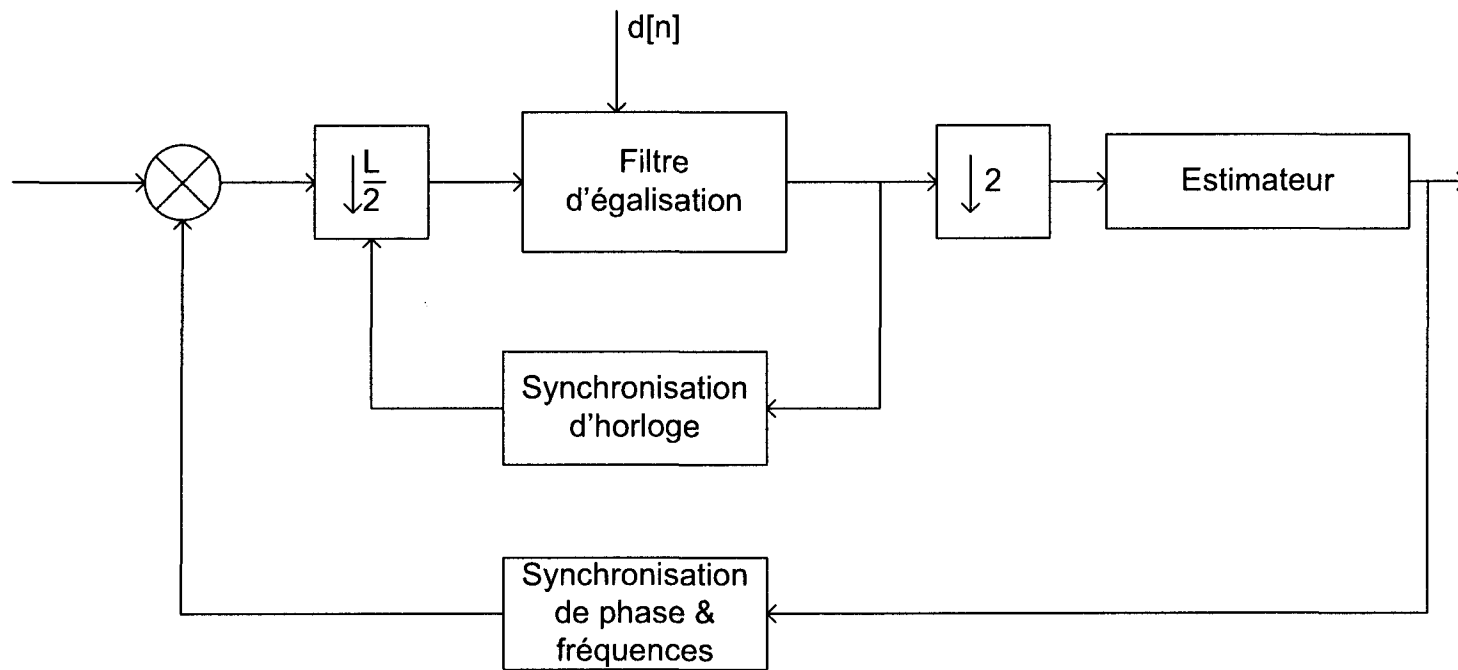
Dans ce chapitre, l'algorithme utilisé pour chacun des blocs sera détaillé. Nous illustrerons aussi quelques résultats obtenus par simulation sur Simulink concernant la performance de certains blocs par rapport aux objectifs de design fixés.

### 3.1 Format des paquets transmis

Avant la transmission dans le canal, les données sont divisées en blocs. Une entête est ajoutée à chaque bloc de données pour former un paquet. Chaque paquet transmis est de



(a) Schéma Bloc du récepteur



(b) Agencement des blocs de synchronisation

FIGURE 3.2 Structure du récepteur modélisé



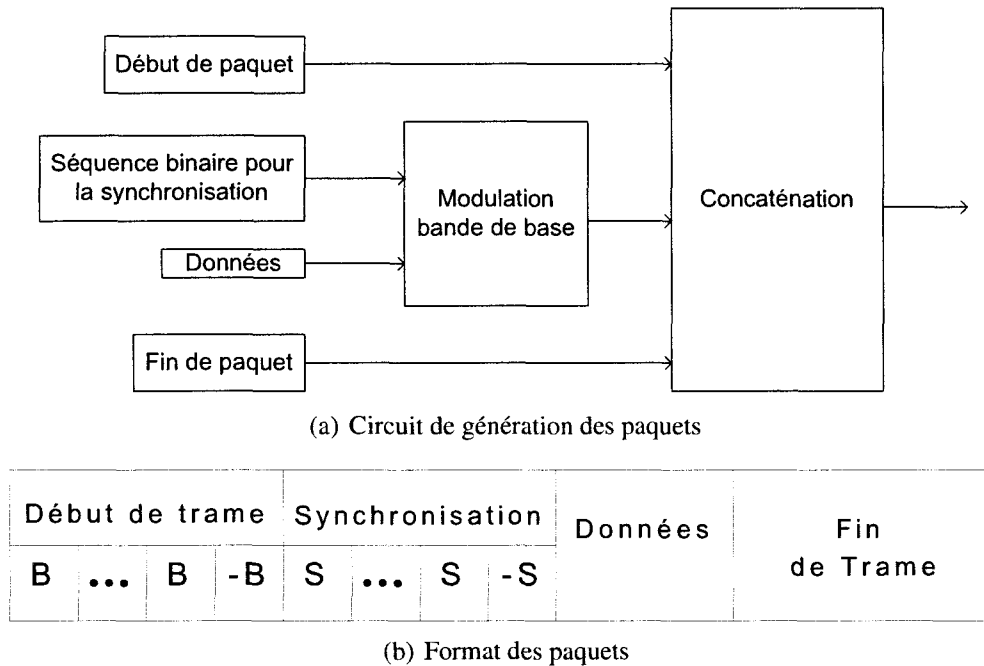


FIGURE 3.3 Vue générale des paquets

la forme illustrée à la figure 3.3. Une séquence de début de paquet forme l'entête. À cela est concaténée la séquence de synchronisation. Cette dernière est générée à partir d'une séquence binaire modulée. Les données et, optionnellement, la séquence de fin de paquet s'ensuivent.

### 3.1.1 Séquence de début des trames

Un code de Barker est une séquence de  $+1$  et de  $-1$  ayant une faible auto-corrélation. Le tableau 3.1 illustre toutes les séquences de Barker connues [Terr]. Nous pouvons observer que l'auto-corrélation des séquences est de 1 au plus lorsque les séquences ne sont pas parfaitement alignées. À noter que par conjecture, il a été démontré qu'il n'existe pas de code de Barker d'une longueur supérieure à 13.

Lors du design, nous avons utilisé la séquence de Barker de longueur 7 pour débiter

Longueur	Code
2	+ - ou + +
3	+ + -
4	+ - + + ou + - - -
5	+ + + - +
7	+ + + - - + -
11	+ + + - - - + - - + -
13	+ + + + + - - + + - + - +

TABLE 3.1 Les séquences de Barker connues.

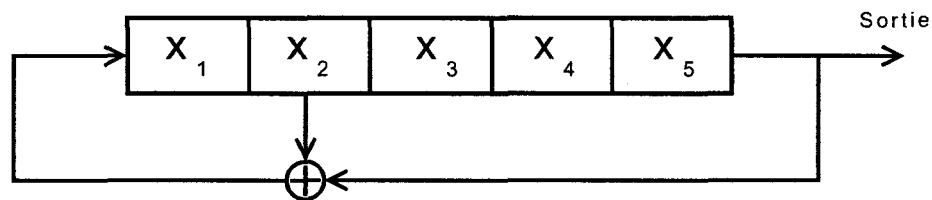


FIGURE 3.4 Génération d'un PRBS avec un registre à décalage.

chaque paquet transmis. Trois séquences, chacune dénotée  $+B$ , sont initialement transmises suivies d'une séquence  $-B$  (la séquence  $B$  déphasée de  $180^\circ$ ). Nous mettrons en évidence les raisons d'une telle structure à la section 3.6. Les séquences de Barker sont transmises dans le canal selon une modulation BPSK de puissance moyenne unitaire.

### 3.1.2 Séquence de synchronisation

Les séquences de synchronisation sont nécessaires afin de permettre au récepteur de s'ajuster aux effets du canal avant la détection des données. Les bits de synchronisation sont constitués à partir d'une séquence binaire pseudo-aléatoire (PRBS),  $S$ , de longueur 31. Celle-ci est générée en utilisant un registre à décalage de longueur 5 avec une rétroaction formée de  $x_2 \oplus x_5$  (c/f figure 3.4). Les registres à décalage sont tous initialisés à 1 et le bit du dernier registre est utilisé pour former la sortie. Cette section du paquet est formée de  $N - 1$  séquences  $S$  suivie d'une séquence  $S$  inversée pour détecter la fin de la période de synchronisation où  $N$  est un nombre pair. Les bits générés sont modulés avant la transmission à travers le canal. Par la suite, les bits de données modulés sont transmis.

### 3.1.3 Séquence de fin des trames

En général, une séquence binaire peut être utilisée pour détecter la fin de la transmission des données. Cependant, nous voulons ici estimer la performance de notre circuit en termes de taux d'erreurs binaires (BER) pour plusieurs valeurs de SNR. Ainsi, nous avons formé des paquets de longueur constante connue au récepteur sans insertion de séquence de fin de paquet. Le récepteur pourra alors déterminer la fin d'un paquet en comptant le nombre de symboles de données reçus.

## 3.2 Le modulateur et le démodulateur

La modulation QPSK est utilisée pour les séquences de synchronisation et les données pour les raisons suivantes :

1. Il est désirable d'effectuer le design des modules d'une telle façon à utiliser les dimensions réelles et imaginaires. Même si une constellation avec plus de points est envisageable, le QPSK permet de faire une analyse rapide de la fonctionnalité globale du système sans ajouter davantage de complexité, excepté au niveau du démodulateur.
2. Elle est une des constellations proposées pour la classe de services de troisième génération [3GPP TS 25.223 v.8.1.0].

L'ordonnancement de Gray est utilisé et la puissance moyenne de transmission est unitaire. Le taux d'erreurs binaires théorique dans un canal AWGN et l'efficacité spectrale se trouvent dans le tableau 1.2.

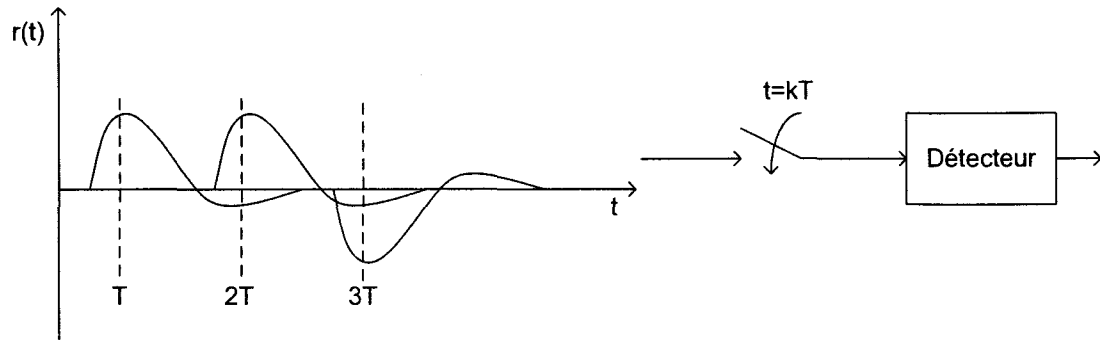


FIGURE 3.5 Interférences entre symboles.

### 3.3 Filtres de Nyquist

Lorsqu'un canal est limité en largeur de bande à  $W_c$  Hz, il peut être modélisé comme un filtre ayant une réponse impulsionnelle  $c(t)$  et une réponse en fréquences  $C(f)$ . Le canal atténue toutes composantes fréquentielles  $|f| < W_c$  du signal et y ajoute un délai de phase propre à chaque fréquence. Les composantes  $|f| > W_c$  sont supprimées. Une signalisation efficace utilise toute la portion de largeur de bande disponible dans le canal [Proakis, J., 2000]. De plus, dû au délai de phase introduit dans le canal, les symboles au récepteur sont intermêlés (Voir figure 3.5). Ceci est le phénomène d'interférences entre symboles (ISI). Le récepteur échantillonne le signal reçu après le filtrage adapté aux instants  $kT$ . À la figure 3.5, pour les échantillons obtenus aux instants  $t = 2T$  et  $t = 3T$ , nous constatons que la valeur de l'échantillon courant est corrompue par l'interférence causée par le symbole précédent.

#### Critère de Nyquist

Nyquist a étudié le problème de la transmission sans ISI dans les canaux à bande limitée. Il en a déduit que pour effectuer la transmission à travers un canal idéal de largeur de bande  $W_c$  sans ISI, le taux de transmission des symboles  $R_s$  doit être inférieur à  $2W_c$  et la réponse impulsionnelle du système complet doit répondre à certains critères particuliers

[Proakis, J., 2000]. Les filtres en cosinus surélevés sont une famille populaire de filtres qui satisfont à ces critères. Leurs définitions sont données par leurs réponses fréquentielles (voir équation 3.1).

$$H_{RC}(f) = \begin{cases} 1 & , \text{ pour } 0 \leq |f| \leq W(1 - \beta), \\ \cos^2\left\{\frac{\pi}{4\beta W}(|f| - W(1 - \beta))\right\} & , \text{ pour } W(1 - \beta) \leq |f| \leq W(1 + \beta), \\ 0 & , \text{ pour } |f| \geq W(1 + \beta). \end{cases} \quad (3.1)$$

La réponse impulsionnelle est

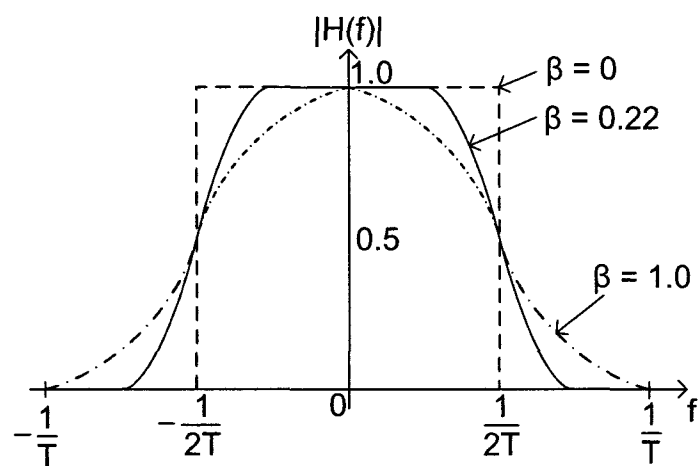
$$h_{RC}(t) = \frac{\cos(2\pi\beta Wt)}{1 - (4W\beta t)^2} \text{sinc}(2Wt) \quad (3.2)$$

où le coefficient de retombée  $\beta$  ( $0 \leq \beta \leq 1$ ) définit la largeur de bande en excès. Une grande valeur de  $\beta$  diminue l'ordre du filtre requis, mais augmente la largeur de bande [Rappaport, T.S., 2002]. La figure 3.6 illustre les caractéristiques de  $X(f)$  et de  $x(t)$  pour différentes valeurs de  $\beta$ . On constate que  $x(kT) = 0 \forall k \neq 0$  quelque soit la valeur de  $\beta$ . Ainsi, nous pouvons en déduire qu'il y aura seulement un symbole  $s_m$  qui influencera le signal reçu à l'instant  $mT + \tau$ ,  $\tau$  étant le délai du canal, ce qui permet la transmission sans ISI si le signal de réception est échantillonné au bon instant.

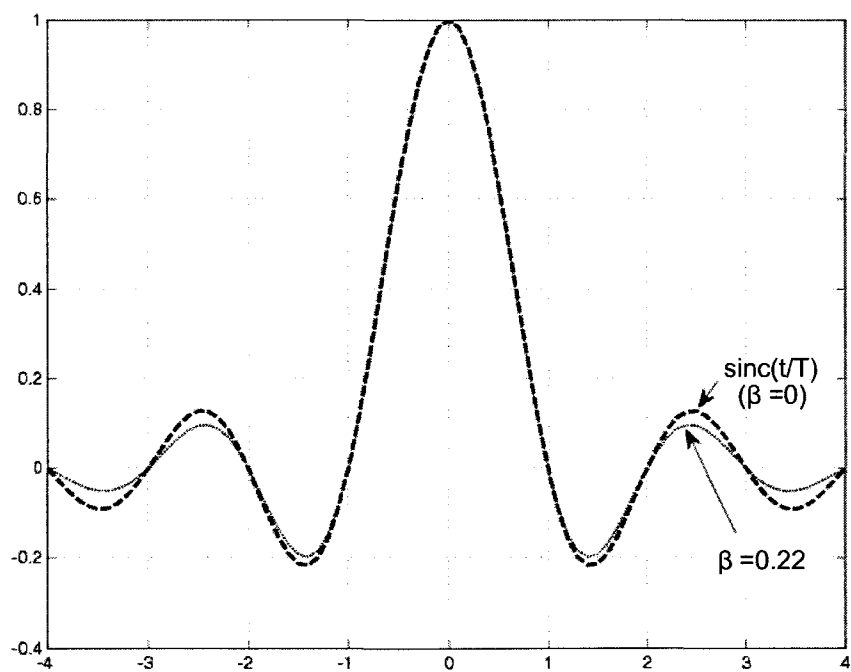
Il est à noter que ces filtres sont irréalisables pour les deux raisons suivantes :

1. La longueur du filtre est infinie pour toutes les valeurs de  $\beta$ .
2. La réponse impulsionnelle est non-causale.

La réponse impulsionnelle devra être de longueur finie et un délai devra être imposé aux symboles pour la rendre réalisable.



(a) Réponse Fréquentielle



(b) Réponse Impulsionnelle

FIGURE 3.6 Réponses des filtres en cosinus surélevés.

### Un filtre pratique

Les filtres en cosinus surélevé étant irréalisables, il est nécessaire de relaxer certaines contraintes de design. De plus, en pratique, deux filtres sont utilisés, soit un filtre à la transmission  $H_T(f)$  permettant de limiter la largeur de bande utilisée et un filtre complémentaire adapté à la réception  $H_R(f) = H_T^*(f)$ . Lors de notre design, nous avons utilisé deux filtres en racine cosinus surélevé. La fonction de transfert de ce filtre est décrite par l'équation 3.3

$$H_{RRC}(f) = \begin{cases} 1 & , \text{ pour } 0 \leq |f| \leq W(1 - \beta), \\ \cos\{\frac{\pi}{4\beta W}(|f| - W(1 - \beta))\} & , \text{ pour } W(1 - \beta) \leq |f| \leq W(1 + \beta), \\ 0 & , \text{ pour } |f| \geq W(1 + \beta). \end{cases} \quad (3.3)$$

La réponse impulsionnelle est alors

$$h_{RRC}(t) = \frac{\sin(\pi \frac{t}{T}(1 - \beta)) + 4\beta \frac{t}{T} \cos(\pi \frac{t}{T}(1 + \beta))}{\pi \frac{t}{T}(1 - (8\beta W t)^2)} \quad (3.4)$$

Pour une réalisation numérique, il est nécessaire d'échantillonner le filtre. Les spécifications de 3GPP suggèrent un taux de suréchantillonnage allant de 4 à 8. Un grand facteur d'échantillonnage permet de relaxer le design analogique mais requiert du traitement plus rapide. Nous avons utilisé un facteur  $L$  fixé à 8 avant de passer à travers le filtre de transmission puisque nous concevons le système pour avoir un taux de transmission faible et des algorithmes qui n'utilisent pas d'interpolation.  $\beta$  est fixé à 0,22 comme utilisé dans le 3GPP. Le gain de chaque tap correspond au coefficient de la réponse impulsionnelle discrétisée de l'équation 3.5.

$$h_{RRC}(n) = \frac{\sin(\pi \frac{n-33}{8}(1 - \beta)) + 4\beta \frac{n-33}{8} \cos(\pi \frac{n-33}{8}(1 + \beta))}{\pi \frac{n-33}{8}(1 - (\frac{\beta(n-33)}{2})^2)} \quad (3.5)$$

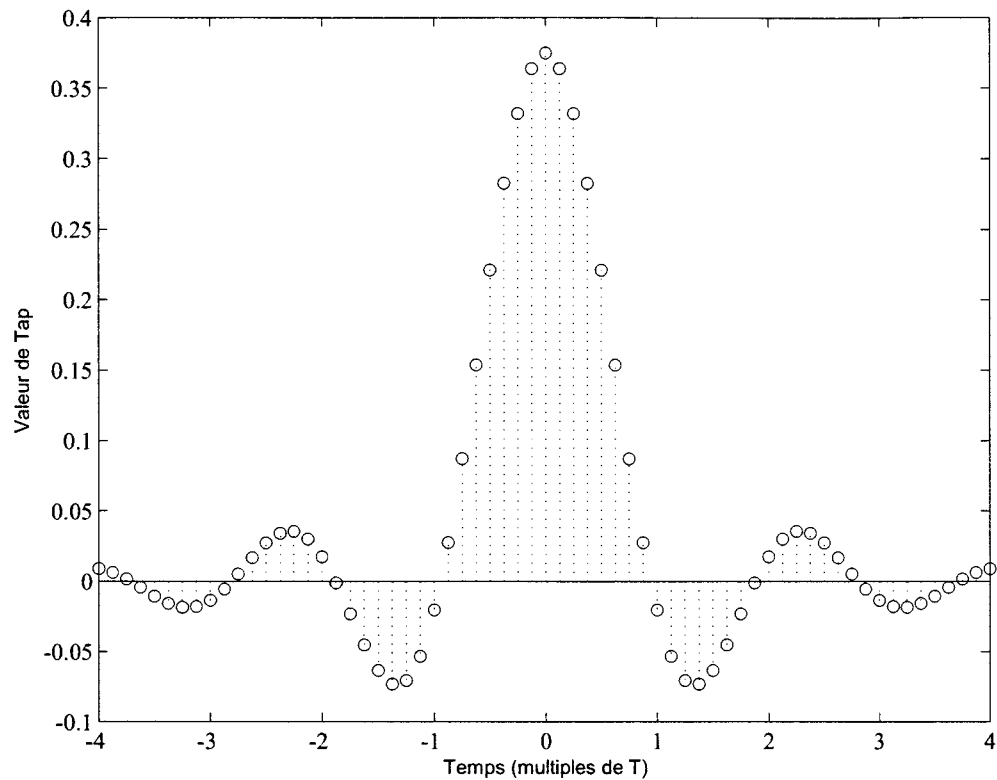


FIGURE 3.7 Réponse impulsionnelle du filtre en racine cosinus surélevé pour mise en forme d'impulsions.

La réponse impulsionnelle du filtre ainsi conçu est illustrée à la figure 3.7

Nous pouvons faire les observations suivantes :

- L'obtention d'un filtre réalisable est au coût d'un délai de groupe et d'une troncation de la réponse impulsionnelle pour  $t < -4T$  et  $t > 4T$ .
- Aux instants  $t=kT$ , le filtre ne chute pas à 0. Ainsi, les filtres en racine cosinus surélevé seuls n'éliminent pas l'ISI. Cependant l'utilisation d'un filtre adapté au récepteur fournit une réponse globale de telle sorte que l'ISI est annulée (c.f figure 3.8).

Pour définir la longueur du filtre à utiliser, nous avons considéré deux facteurs. Dans un premier temps, nous avons simplifié le scénario en utilisant une fenêtre rectangulaire. L'application d'une autre fenêtre, tel qu'une fenêtre de Hamming, est aussi envisageable afin de réduire les lobes secondaires et ainsi réduire davantage l'interférence et le bruit hors bande. Par la suite, nous avons imposé une perte maximale de 0.1 dB pour un SNR



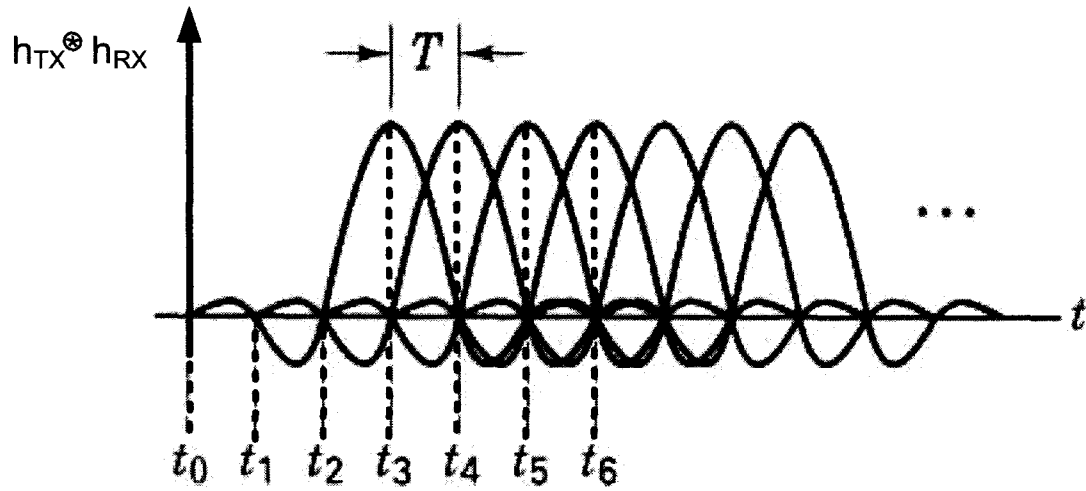


FIGURE 3.8 Symboles à la sortie du filtre adapté.

de 10 dB lorsque le système est simulé dans un canal AWGN. En utilisant des résultats de simulation, nous avons décidé que 65 taps pour les filtres en racine cosinus surélevé au transmetteur et au récepteur satisfaisaient notre critère.

### 3.4 Filtre d'égalisation

Les filtres de Nyquist décrits dans la section précédente permettent de garantir une transmission sans ISI dans un canal à bande limitée. Cependant, dans certaines situations telles qu'en présence de composantes multi-trajets, l'utilisation d'un filtre d'égalisation est nécessaire pour réadapter le signal de réception afin de minimiser la probabilité d'erreurs [Qureshi, 1985]. Pour un filtre linéaire, la sortie  $z[n]$  est obtenue à partir du signal de réception  $y[n]$  par l'équation :

$$z[n] = \sum_{i=-L}^L w_i y[n - i] \quad (3.6)$$

où  $w[i]$  est le gain du  $i^{\text{ème}}$  tap d'un filtre à  $N = 2L + 1$  taps. Nous pouvons aussi écrire

l'équation de la fonction de transfert  $\hat{H}_{eq}(z)$  :

$$\hat{H}_{eq}(z) = \sum_{i=-L}^L w_i z^{-i} \quad (3.7)$$

Le nombre de taps à utiliser est typiquement relié à des questions d'implémentation et au canal à égaliser. Avoir un grand nombre de taps implique un plus grand niveau de complexité pour une meilleure résolution du canal. Le travail consiste ici à déterminer les valeurs  $w_i$  des coefficients des taps du filtre (entraînement) et les mettre à jour (tracking).

Nous avons utilisé un filtre LMS (Least Mean Square) qui adapte ses coefficients de sorte à minimiser l'erreur quadratique moyenne. À chaque itération, l'algorithme de réajustement de coefficients est

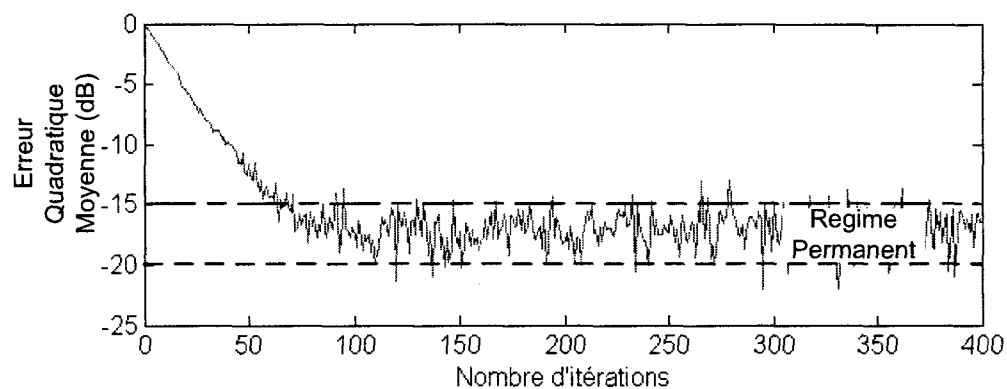
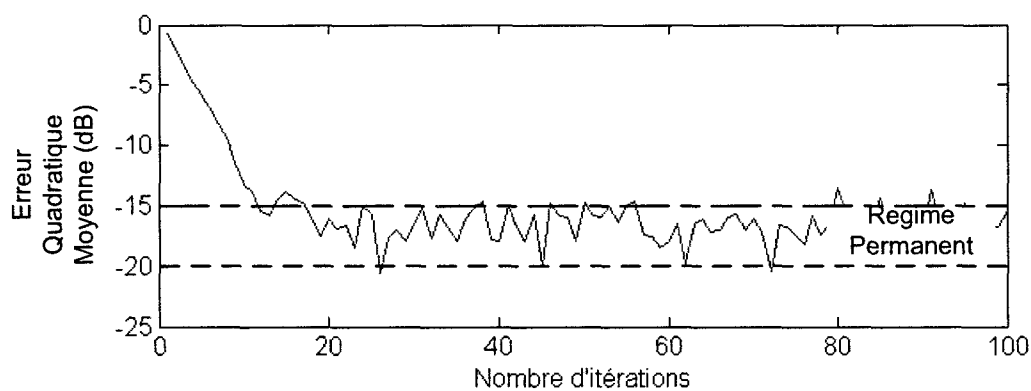
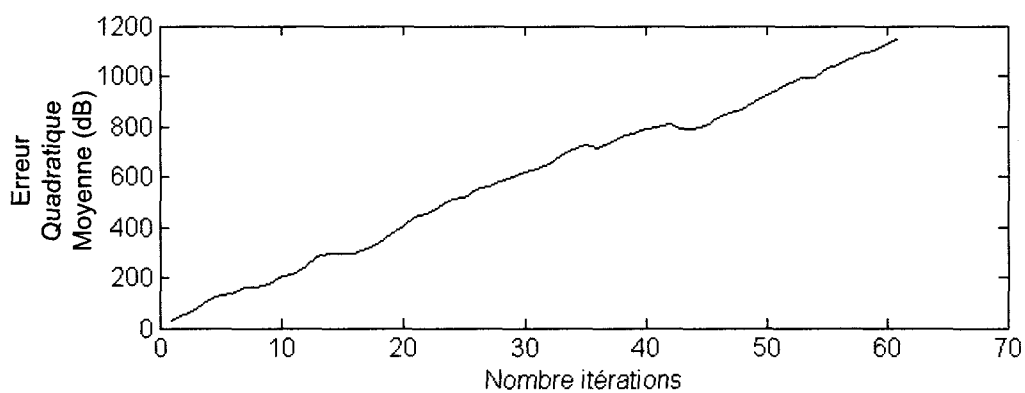
$$\overrightarrow{W}_{n+1} = \overrightarrow{W}_n + \mu \epsilon[n] \overrightarrow{y^*[n]} \quad (3.8)$$

où  $\epsilon[n] = z[n] - d[n]$  est l'erreur entre la sortie du filtre et la séquence d'entraînement  $d[n]$ ,  $\mu$  est le pas à chaque itération,  $\overrightarrow{y^*[n]} = [y^*[n-L] \cdots y^*[n] \cdots y^*[n+L]]$  et  $\overrightarrow{W}_n = [w_{-L}[n] \cdots w_0[n] \cdots w_L[n]]$ . La valeur de  $\mu$  doit être choisie de sorte à avoir un bon compromis entre la vitesse de convergence et la stabilité du filtre [Widrow, B., 1985]. Une grande valeur de  $\mu$  permet d'effectuer des grands pas à chaque itération pour converger plus rapidement à une solution mais peut rendre le filtre instable.

De plus, en effectuant une moyenne sur  $\epsilon^*[n]\epsilon[n]$  lorsque le filtre a convergé vers sa solution finale, nous pouvons obtenir l'erreur quadratique moyenne minimale (MMSE pour Minimum Mean Square Error). À noter que le SNR de réception peut être obtenu à partir de l'équation 3.9 (voir la section I.3).

$$SNR = \frac{P_t}{MMSE} - 1 \quad (3.9)$$



(a)  $\mu = 0.01$ (b)  $\mu = 0.05$ (c)  $\mu = 1.5$ FIGURE 3.10 Taux de convergence de l'algorithme LMS pour différentes valeurs de  $\mu$ .

### 3.5 Circuit d'ajustement initial de puissance

Puisque le signal est atténué à travers le canal, il est nécessaire de procéder à un contrôle de gain de puissance avant de numériser et de traiter le signal de réception. Ceci a pour but d'optimiser l'utilisation de la plage dynamique des convertisseurs analogique-numérique. Lors de la conception du contrôleur de gain automatique (AGC), nous avons fixé les objectifs suivants :

#### Plage de gain du contrôleur

Il est évident que le design doit prendre en considération l'atténuation qui peut être subie dans le canal. En se basant sur l'hypothèse que le canal atténue le signal de l'ordre de 60 dB dans le pire cas, nous avons conçu un AGC ayant un gain variable au plus de 70 dB.

#### Temps de réponse

En pratique, le contrôleur de gain doit être vu comme un estimateur du niveau de puissance du signal de réception [Proakis, J., 2000]. Donc, la compensation du gain se fait selon une moyenne de la puissance sur un intervalle relatif au taux de signalisation plutôt que sur une réponse basée sur le niveau du signal instantané. Nous avons conçu l'AGC de sorte à avoir un niveau de signal de l'ordre de 1.5 dB par rapport au niveau de signal désiré dans un intervalle de 10 durées de symboles à partir de la pire condition du canal, c'est-à-dire une atténuation de 60 dB.

#### 3.5.1 Circuit utilisé

Le contrôleur de gain peut être perçu comme un circuit d'estimation et d'inversion du canal. En prenant en compte l'hypothèse de canal invariant, nous l'avons conçu de sorte à ce que l'ajustement de gain se fasse durant la phase d'entraînement du récepteur. Par la suite, les données sont amplifiées par le même facteur de gain de puissance  $G_{AGC}$ . La

figure 3.11(a) illustre le circuit utilisé.

Dans un premier temps, l'amplitude du signal  $s(t)$  est estimée. Comme mentionné précédemment, nous nous intéressons à une valeur moyenne de la puissance du signal. À cette fin, le filtre passe-bas permet d'atténuer les variations rapides du signal. Le filtre utilisé est décrit par l'équation 3.10 et est d'ordre 1. La valeur de  $\alpha_{AGC}$  détermine l'influence des nouvelles données sur la sortie du filtre. En effet, lorsque  $1 - \alpha_{AGC}$  est petit, l'entrée  $x[n]$  a peu d'influence sur  $\delta[n]$ , donc la sortie du filtre sera moins sensible aux fluctuations bruitées de l'entrée mais convergera plus lentement.

$$\delta[n] = (1 - \alpha_{AGC})x[n] + \alpha_{AGC}\delta[n - 1] \quad \text{avec } 0 < \alpha_{AGC} < 1 \quad (3.10)$$

La comparaison subséquente des données avec un seuil de référence fournit le niveau de rajustement de puissance  $\delta$  nécessaire. Les deux circuits des figures 3.11(b) et 3.11(c) sont alors utilisés pour déterminer le gain à appliquer.

Le premier circuit ("Coarse") fait un premier réajustement de la tension. Ce réajustement est nécessaire pour permettre une convergence initiale rapide vers la solution. L'incrément du niveau de gain  $\Delta G_1$  généré par ce circuit est décrit selon l'équation 3.11.

$$\Delta G_1 = \begin{cases} 0.5 & \text{pour } \delta > 2 \\ -0.5 & \text{pour } \delta < -2 \\ 0 & \text{sinon} \end{cases} \quad (3.11)$$

Le gain  $G_{coarse}$  est donc

$$G_{coarse} = 10^{\sum_{i=0}^n \Delta G_1[i]} \quad (3.12)$$

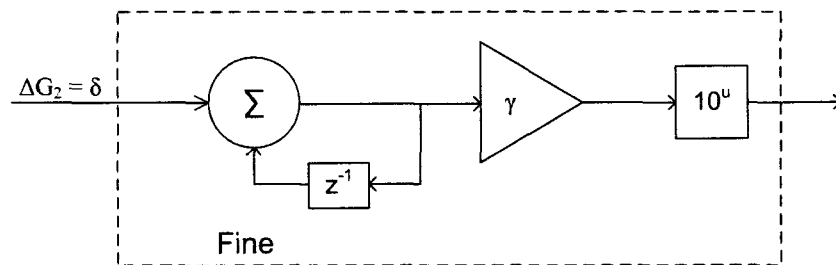
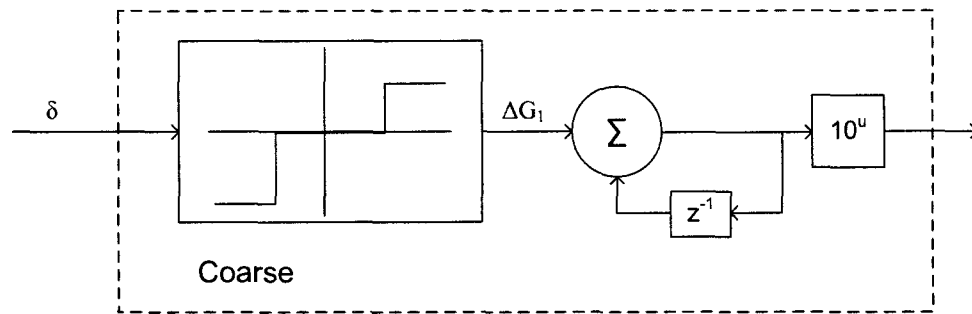
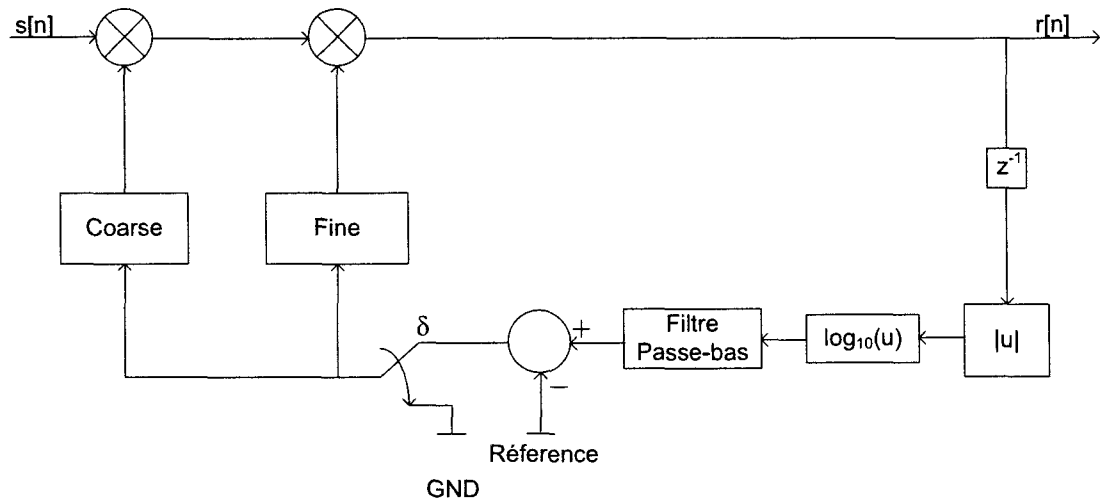


FIGURE 3.11 Contrôleur de gain automatique.

Un deuxième circuit ("Fine") opère en parallèle pour produire un ajustement plus raffiné du gain  $G_{fine}$  nécessaire. Nous avons ici conçu le circuit de façon à ce que l'incrément du niveau de gain se fasse selon une fonction continue de  $\gamma$ .

$$G_{fine} = 10^{\gamma \sum_{i=0}^n \delta[i]} \quad (3.13)$$

Le facteur  $\gamma$  permet d'ajuster le taux de convergence. Suite à l'entraînement du système, les entrées des intégrateurs sont forcées à 0 de sorte que le gain global de l'AGC ne change plus.

### 3.5.2 Tests de performance

Les performances de l'AGC ont été mesurées par sa vitesse de convergence pour différentes atténuations du canal. Nous avons considéré un canal non-bruité. À la figure 3.12, nous montrons les résultats obtenus pour une amplification de 6dB du signal ainsi que pour des atténuations de 0 dB, 30 dB et 60 dB. Nous avons considéré que la convergence est atteinte lorsque le signal de sortie est entre -1 dB et 1 dB.

Le nombre d'itérations correspond ici à la quantité de données fournie à l'AGC, soit le produit du nombre de symboles par le taux de suréchantillonnage. Pour une atténuation de 60dB, nous observons que le temps de convergence est d'environ 50 à 60 échantillons. Le taux de suréchantillonnage que nous avons utilisé est de 8, le système a donc nécessité moins d'une dizaine de symboles pour converger.

## 3.6 Détection de début de trame de synchronisation

En référence à la figure 3.2(a), les circuits de synchronisation sont déclenchés suite à la détection de la séquence de début de trame séquence  $\{+B, +B, +B, -B\}$ .  $B$  se réfère ici à



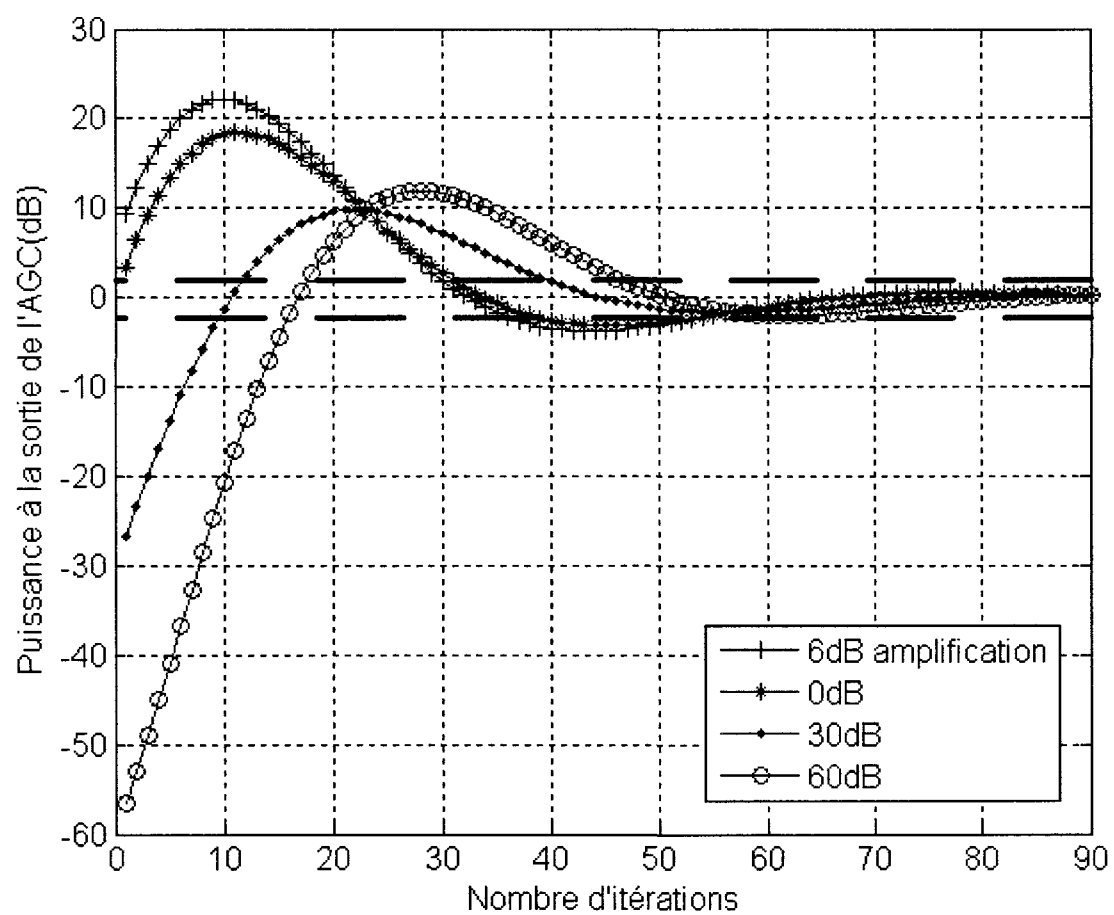


FIGURE 3.12 Résultat des tests effectués sur le contrôleur de puissance.

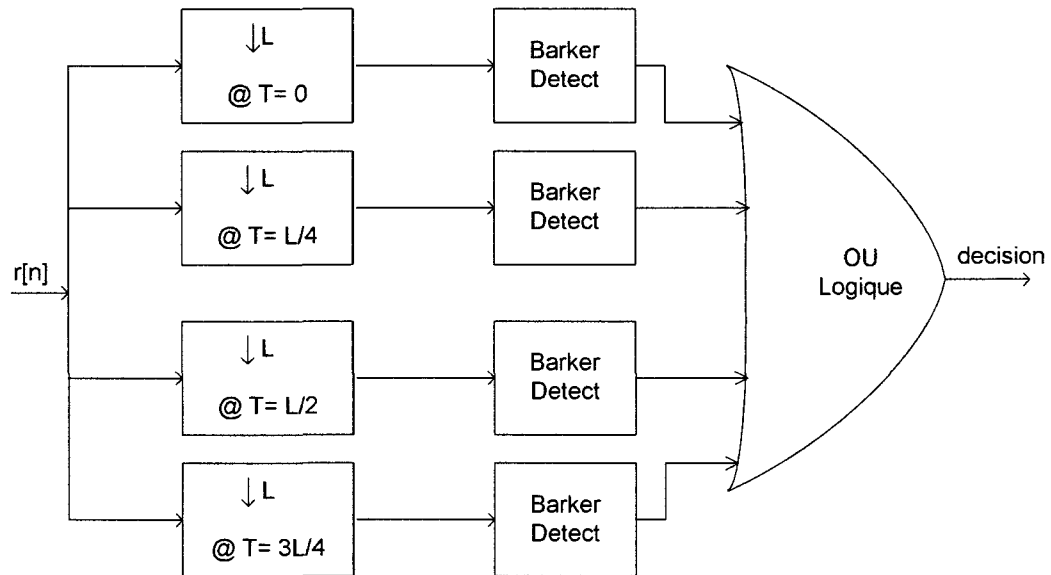


FIGURE 3.13 Circuit de détection de début de trame.

une séquence de Barker de longueur  $N$  (voir la section 3.1.1). À l'exception de l'AGC, l'apprentissage du canal n'a ici pas encore été effectué, donc la détection doit s'effectuer sans connaissance de la phase du signal de réception.

Nous proposons la détection de la séquence selon les composantes en phase et en quadrature du signal de réception à quatre instants d'échantillonnages. Donc le circuit de la figure 3.13 est répété pour les composantes en phase et quadrature. Le début d'une séquence d'entraînement est déclaré lorsqu'il est détecté par un des quatre sous-blocs identiques dénotés *Barker Detect*. Le circuit de chaque bloc *Barker Detect* est détaillé à la figure 3.14.

$N$  est la longueur de la séquence de Barker utilisée. Nous effectuons initialement une corrélation sur le signe des  $N$  échantillons du signal de réception avec la séquence de Barker utilisée (et connue au récepteur). En utilisant un seuil convenable de détection, nous sommes en mesure de décider si une séquence de Barker a été détectée. Pour détecter le début de la séquence de synchronisation subséquente, nous prenons avantage du changement de signe imposé à la dernière séquence de début de chaque trame. Lorsque

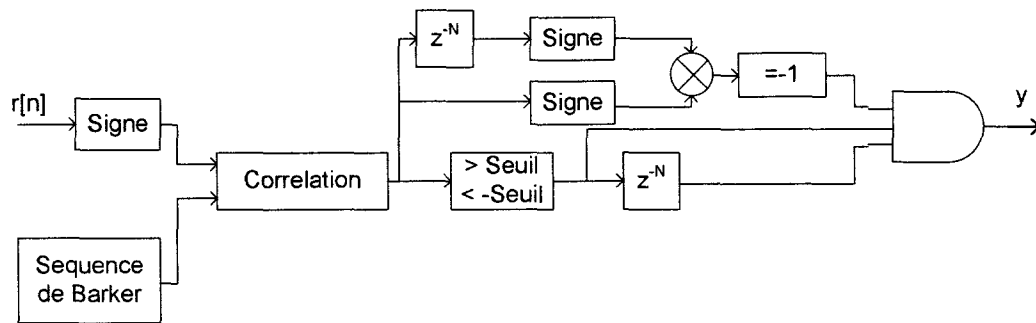


FIGURE 3.14 Circuit de détection de séquence de Barker.

deux séquences de Barker sont détectées à une séparation de  $N$  échantillons avec changement de signe, nous décidons que la séquence de synchronisation commencent et le reste du récepteur est alors activé.

Pour évaluer la performance, nous avons évalué le taux de pertes de paquets pour différents niveaux de SNR. Un paquet est perdu lorsqu'il n'est pas détecté. La figure 3.15 illustre le taux de pertes de paquets simulés pour différents niveaux de SNR lorsque le seuil de détection est fixé à un niveau de corrélation de 6. Nous observons que pour un rapport signal à bruit supérieur à 7 dB, le taux de pertes de paquets est inférieur à 1 %.

### 3.7 Synchronisation du récepteur

L'acquisition précise du temps d'échantillonnage des symboles et de la phase est essentiel pour tout démodulateur cohérent. Plusieurs techniques ont été développées au courant des années récentes dont des circuits en boucle ouverte et en boucle fermée [Sklar, B., 2001].

En général, l'estimation de la phase et du temps d'échantillonnage des symboles revient à estimer un paramètre du signal en présence de bruit. La présence de multi-trajets et d'un canal à évanouissement rapide ajoute un défi supplémentaire à la tâche [Goldsmith A., 2005]. Toutefois, le travail revient à effectuer l'estimation d'un certain paramètre suivi d'un algorithme de compensation ou de correction.

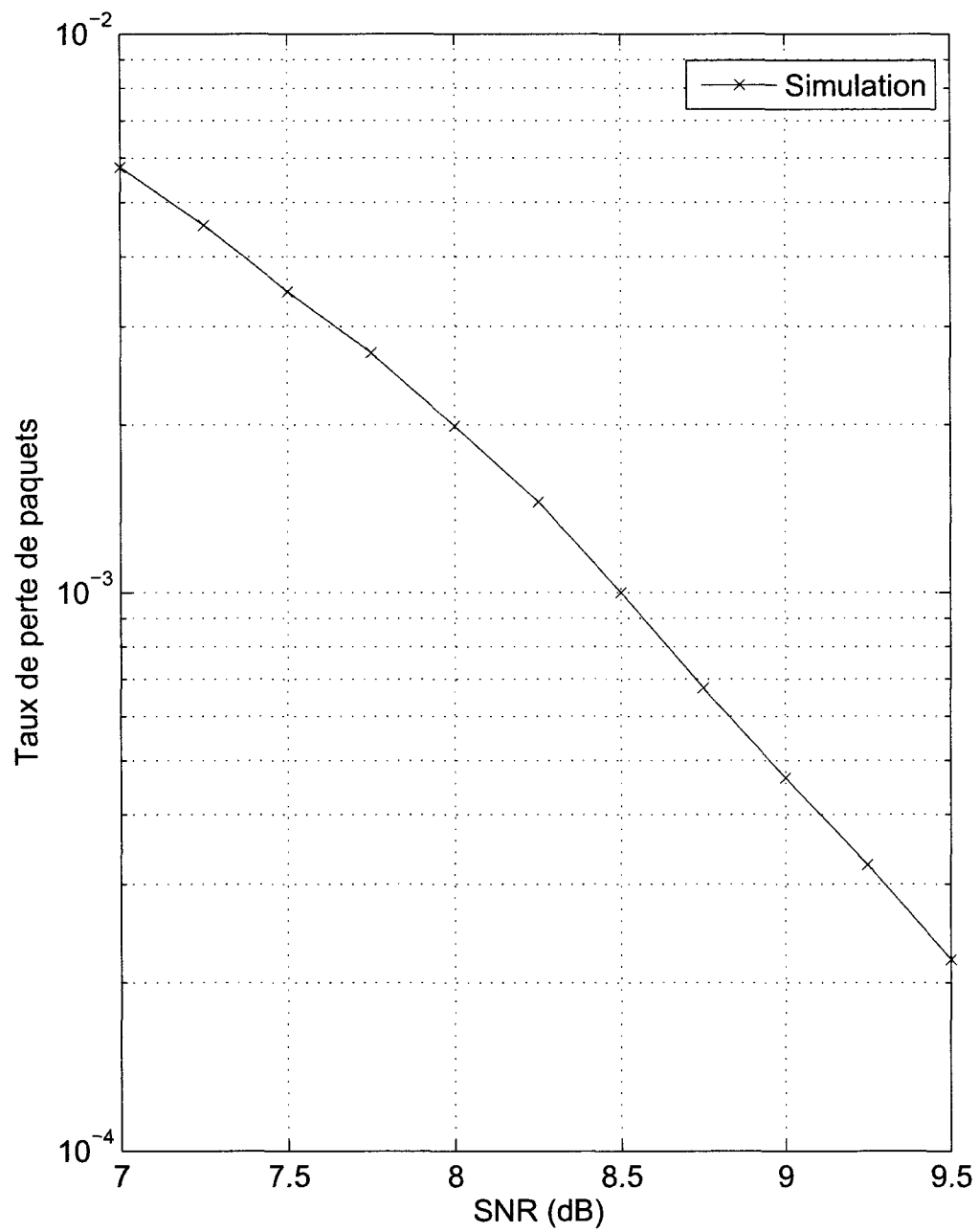


FIGURE 3.15 Taux de pertes de paquets pour différents niveaux de SNR avec un seuil de détection à 6

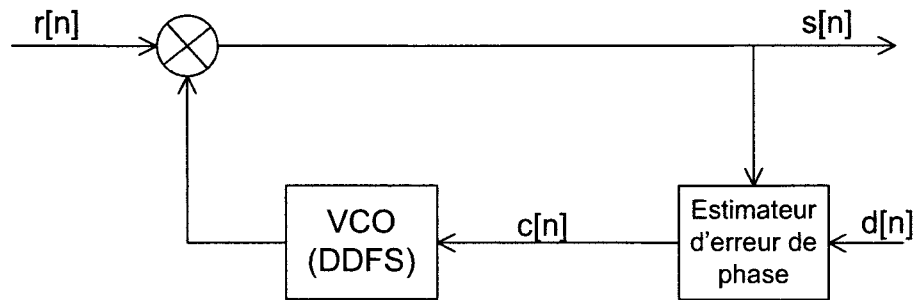


FIGURE 3.16 Schéma bloc de la boucle de correction de phase.

### 3.7.1 Synchronisation de phase et de fréquence

Le Phase Locked Loop (PLL) est le système de contrôle qui se trouve au coeur de l'ajustement de la phase et de la fréquence dans la majorité des systèmes de communications [Sklar, B., 2001]. Nous avons divisé la conception de notre système en deux blocs. Le premier bloc effectue une estimation de l'erreur de phase du signal de réception par rapport à un signal de référence. En se basant sur cette erreur de phase, le deuxième bloc génère une sinusoïde pour corriger le signal de réception.

Sans perte de généralité, nous pouvons considérer qu'une différence de fréquence revient à une variation temporelle du déphasage du signal [Meyr, H., Ascheid, G., 1997]. Ceci provient du fait que la phase est la dérivée temporelle de la fréquence. Nous pouvons donc considérer que l'ajustement fréquentiel se fera de facto.

#### Estimation de phase

Le signal de réception  $r(n)$  doit être ajusté par rapport au signal de référence  $d(n)$ . En considérant la forme angulaire de ces signaux, et en effectuant les opérations de l'équation 3.14, nous arrivons à avoir une estimation de l'erreur de phase.

$$\begin{aligned}
e_{phase}[n] &= s[n]d^*[n] \\
&= e^{j\phi} (e^{j\phi_d})^* \\
&= e^{j(\phi - \phi_d)} \\
&= \cos(\phi - \phi_d) + j\sin(\phi - \phi_d)
\end{aligned} \tag{3.14}$$

À la figure 3.17, nous constatons que cette opération est une projection de  $s(t)$  sur  $d(t)$  et elle correspond à une rotation de  $s(t)$  d'un angle  $-\phi_d$  dans le plan complexe. Nous pouvons aussi faire les observations suivantes :

- Si le signe de la partie imaginaire de  $e_{phase}[n]$  est positif, alors  $\phi > \phi_d$  et la phase du signal de réception est en avance par rapport à la référence.
- Si le signe de la partie imaginaire de  $e_{phase}[n]$  est négatif, alors  $\phi < \phi_d$  et la phase du signal de réception est en retard par rapport à la référence.
- Si la partie imaginaire de  $e_{phase}[n]$  est nulle, le signal est parfaitement synchronisé en phase avec la référence.

En se basant sur ces observations, nous avons conçu le détecteur de phase de la figure 3.18. Le signe de la partie imaginaire de l'erreur détermine donc s'il faut augmenter ou diminuer la phase. Un filtre passe-bas (LPF) est inséré pour éliminer les variations causées par le bruit additif. Le filtre utilisé à la figure 3.16 est d'ordre 1. Le gain de la boucle de rétroaction définit la fréquence de coupure ainsi que le temps de convergence du système de façon connexe au filtre utilisé à la section 3.5. Le gain  $K_{PLL}$  permet de définir la plage de fréquences qui pourront être corrigées par le VCO.

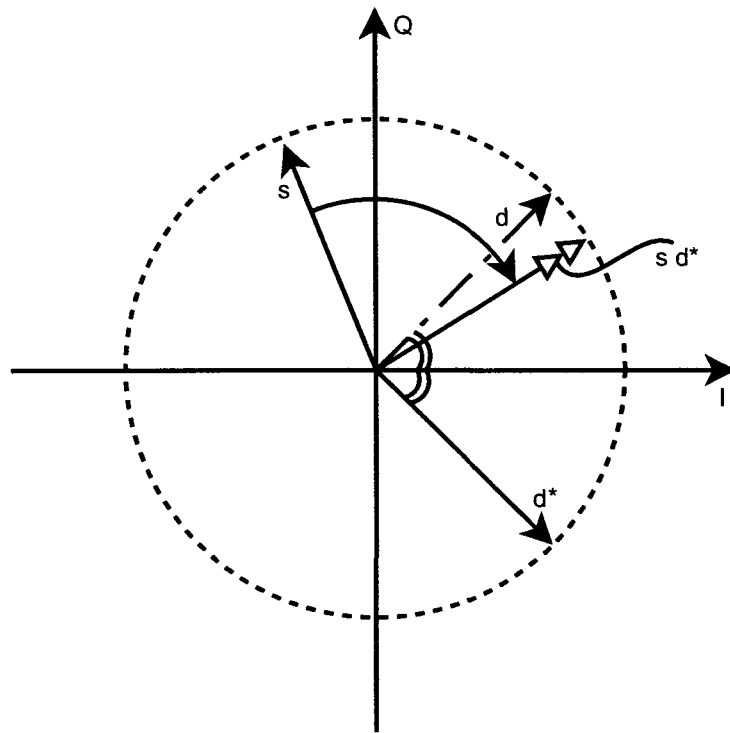


FIGURE 3.17 Comparaison de phases du signal de réception et de la référence dans le plan I-Q

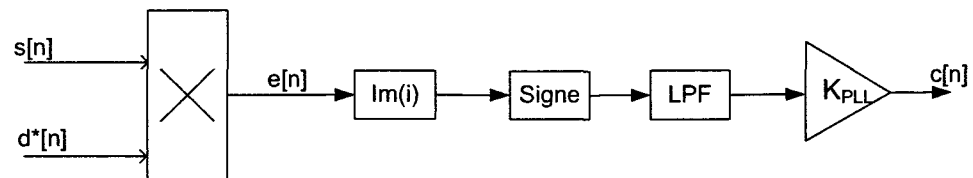


FIGURE 3.18 Estimation d'erreur de phase.

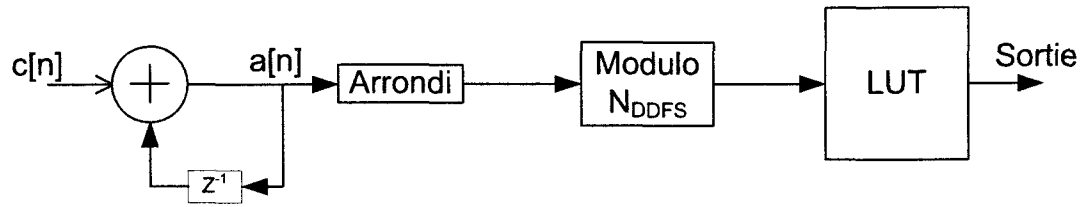


FIGURE 3.19 Circuit d'un DDS.

### Correction de phase et de fréquence

Le DDS (Direct Digital Frequency Synthesizer) de la figure 3.16 a pour fonction d'osciller à une fréquence en se basant sur la tension indiquée par  $c[n]$ . Un circuit intégrateur placé à l'entrée du DDS effectue une accumulation de la correction requise indiquée par  $c[n]$ . Le LUT (Look-up table) est un tableau où nous retrouvons les phases possibles d'un sinusoïde. La longueur du tableau détermine la résolution des phases. Nous avons défini les données du tableau pour une longueur de tableau  $N_{DDFS}$  selon la formule 3.15.

$$Y[i] = e^{-\frac{j \cdot 2\pi i}{N_{DDFS}}}, \text{ pour } i = 0 \cdots (N_{DDFS} - 1) \quad (3.15)$$

Donc, la fréquence générée par le DDS est proportionnelle à la tension  $c[n]$  à son entrée. Lorsque cette tension est nulle, la sortie de l'intégrateur n'incrémente pas. Par contre, pour d'autres valeurs de  $c[n]$ , l'intégrateur incrémente à chaque itération. Ceci fait en sorte qu'à chacune des itérations, différentes valeurs de la table sont choisies. Donc, une sinusoïde est ainsi générée.

### 3.7.2 Recouvrement d'horloge

Jusqu'à présent, nous avons assumé que le récepteur effectue l'échantillonnage du signal à exactement le bon instant de sorte à ce que l'échantillon correspond au symbole transmis à travers le canal. Cependant, en pratique, le récepteur n'a pas de connaissance de l'horloge



du transmetteur. Ceci cause clairement une dégradation des performances du système. Par exemple, l'ISI ne peut pas être minimisée puisque les filtres de Nyquist requièrent l'échantillonnage au bon instant. Dans certains cas, une ambiguïté temporelle de l'ordre du nanoseconde entre les références temporelles du transmetteur et du récepteur peut rendre la récupération de l'information transmise impossible [Stephens, D.R., 2001]. Par exemple, les systèmes à haute largeur de bande tel que le UWB, sont très sensibles au décalage temporel.

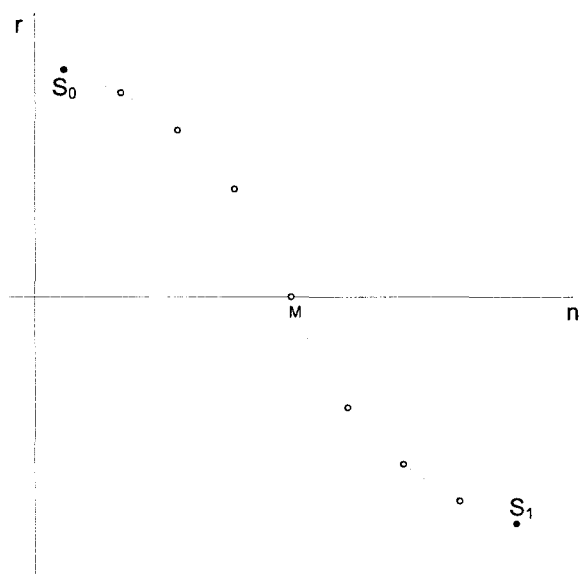
Plusieurs techniques de synchronisation d'horloge [Mueller, K.H. & Muller, M., 1976] ont été proposées dans la littérature. La technique suivante utilise le changement de signe sur la partie réelle et/ou imaginaire du signal de réception pour déterminer si l'échantillonnage se fait en avance, à temps, ou en retard par rapport au bon moment. Les scénarios possibles sont illustrés de façon simplifiée à la figure 3.20.

À des fins d'illustration, considérons deux symboles BPSK suréchantillonnés par un facteur de 8 et transmis à travers les filtres de Nyquist. Le bruit peut être ignoré. Il existe trois scénarios possibles. Le niveau de signal peut être en croissance, en décroissance ou ne pas changer de signe.

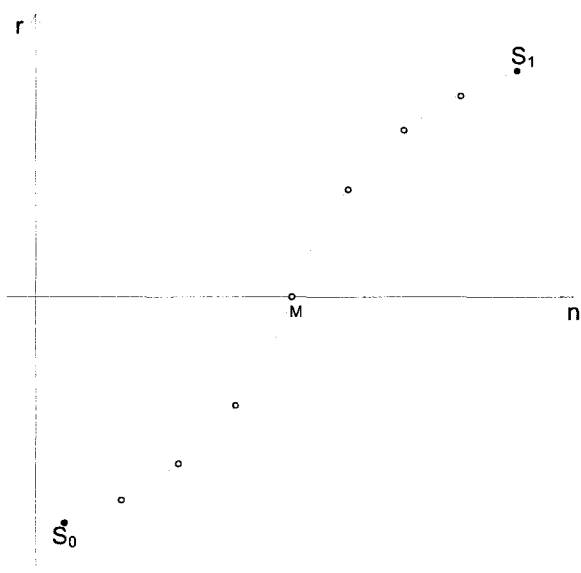
En référence à la figure 3.20(a), nous pouvons faire les observations suivantes pour une transition de positif à négatif :

1. Si le signal est échantillonné en avance par rapport au bon instant, l'échantillon du milieu  $M_a$  aura une valeur négative.
2. Si le signal est échantillonné en retard par rapport au bon instant, l'échantillon du milieu  $M_r$  aura une valeur positive.
3. Si l'échantillonnage s'effectue au bon instant, l'échantillon du milieu  $M$  aura une valeur nulle.

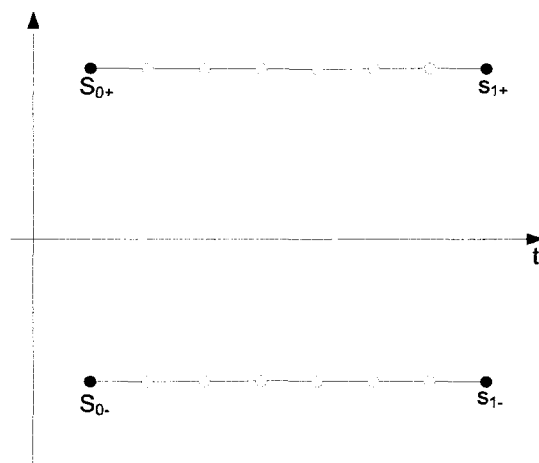
De plus, nous pouvons faire les observations suivantes pour une transition de négatif à positif à partir de la figure 3.20(b) :



(a) Transition en décroissance.



(b) Transition en croissance



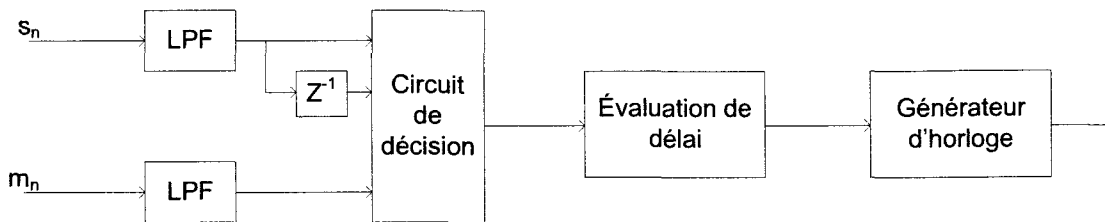


FIGURE 3.21 Circuit de recouvrement d'horloge

1. Si le signal est échantillonné en avance par rapport au bon instant, l'échantillon du milieu  $M_a$  aura une valeur positive.
2. Si le signal est échantillonné en retard par rapport au bon instant, l'échantillon du milieu  $M_r$  aura une valeur négative.
3. Si l'échantillonnage s'effectue au bon instant, l'échantillon du milieu  $M$  aura une valeur nulle.

Cependant, lorsqu'il n'y a pas de changement de signe entre deux symboles consécutives, nous ne pouvons rien conclure sur l'exactitude de l'instant d'échantillonnage.

En se basant sur ces observations, nous avons élaboré le circuit de la figure 3.21. Le circuit de décision se base sur l'échantillon précédent  $s_{n-1}$ , l'échantillon milieu  $m_n$  et l'échantillon courant  $s_n$  pour déterminer s'il faut faire avancer ou reculer l'instant d'échantillonnage. Cet instant d'échantillonnage permet de sélectionner alors le bon échantillon dans la prochaine séquence de  $L$  échantillons,  $L = 8$  étant le facteur de suréchantillonnage. Le bloc d'évaluation de délai décide alors du délai à imposer entre deux coups d'horloge de symbole (horloge pour choisir le bon symbole) et l'indique au générateur d'horloge. Pour illustrer cette dynamique, considérons l'exemple suivant.

### Exemple de la dynamique du temps d'échantillonnage

Considérons la figure 3.22 où chaque colonne correspond à un aux échantillons reliés à un symbole. Le signe  $-$  correspond à un échantillon de signe négatif et le signe  $+$  corres-

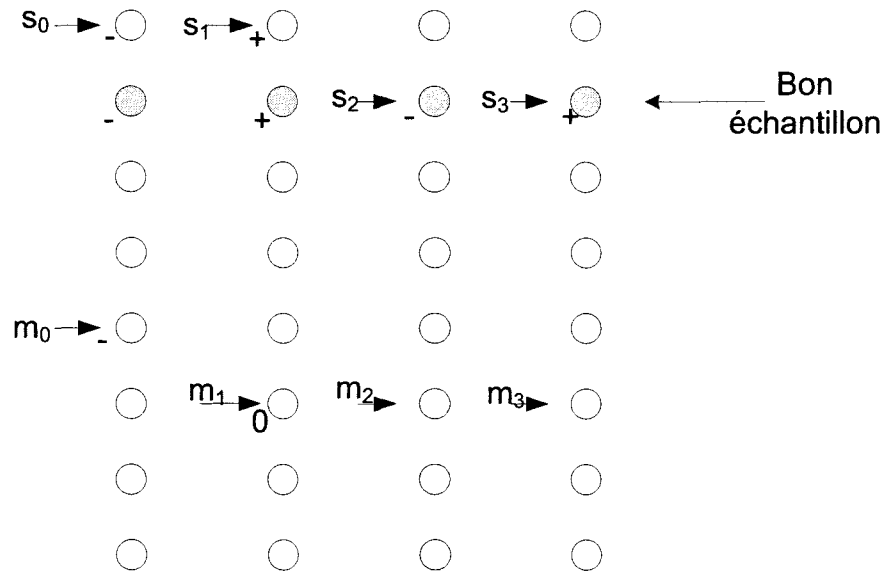


FIGURE 3.22 Exemple de synchronisation d'horloge

pond à un échantillon de signe positif. Le bon échantillon est le deuxième échantillon de chaque colonne et les échantillons choisis sont indiqués par les flèches. Nous avons défini un groupe d'échantillons comme étant les 8 échantillons associés à un symbole. Dans cette section, un groupe de quatre échantillons se réfère aux 4 premiers ou 4 derniers échantillons suivant un symbole. En référence à la figure 3.22, considérons la situation où l'échantillon choisi du premier groupe de quatre échantillons est  $s_0$ . Ce dernier est en retard par rapport au bon échantillon. L'échantillon du milieu  $M_0$  est négatif, et nous observons une transition de négatif ( $s_0$ ) à positif ( $s_1$ ). Le circuit de décision indique qu'il faut donc faire avancer l'instant d'échantillonnage par rapport à l'instant courant. Cela signifie qu'il faudra choisir le deuxième échantillon dans chaque fenêtre de quatre échantillons. À ce moment, le bloc d'évaluation de délai demande au générateur d'horloge de générer le prochain coup à un intervalle de retard, et par la suite de retourner à un intervalle de quatre échantillons. Les deux prochains échantillons choisis sont alors  $m_1$  ayant une valeur nulle et le bon échantillon  $s_2$ . La transition perçue est de positif à négatif et le symbole de milieu est nul. De cette façon, les symboles subséquents choisis seront les bons.

### Situations particulières

À noter les deux situations particulières suivantes dont le bloc d'évaluation de délai doit tenir en compte.

1. Lorsque le symbole courant choisi est le 1<sup>er</sup> du groupe et que le prochain symbole est le 8<sup>me</sup>, le sélecteur de symbole devra sélectionner deux échantillons dans le même groupe de 8 échantillons correspondant à un même symbole dont le premier échantillon et le troisième. En effet, un tel scénario peut se produire lorsque l'échantillon courant se trouve au début du groupe et en retard par rapport au bon échantillon .
2. Lorsque le symbole courant choisi est le 8<sup>me</sup> du groupe et que le prochain symbole est le 1<sup>er</sup>, le sélecteur de symbole devra ignorer le prochain groupe de 8 échantillons correspondant à un symbole et choisira le premier échantillon de celui d'après. Ce scénario peut se produire lorsque l'échantillon courant se trouve à la fin du groupe mais en avance par rapport au bon échantillon.

### Impact d'un décalage d'un demi échantillon

Les décisions prises sont faites par rapport aux échantillons collectés sans interpolation. Cela implique qu'une des hypothèses posées est d'avoir un délai multiple entier de la période d'échantillonnage entre le transmetteur et le récepteur . Ceci n'est pas nécessairement le cas. À la figure 3.23, nous illustrons le diagramme d'oeil de la partie réelle du signal obtenu dans le cas où le délai est à la moitié de la période d'échantillonnage. Nous considérons que le bruit est absent. Le récepteur réadapte son instant d'échantillonnage mais celui-ci oscillera toujours autour du bon instant. Nous observons que le diagramme d'oeil est constitué de deux diagrammes d'oeil superposés. Le diagramme d'oeil en pointillé est le résultat de l'échantillonnage en retard par rapport au bon instant alors que celui

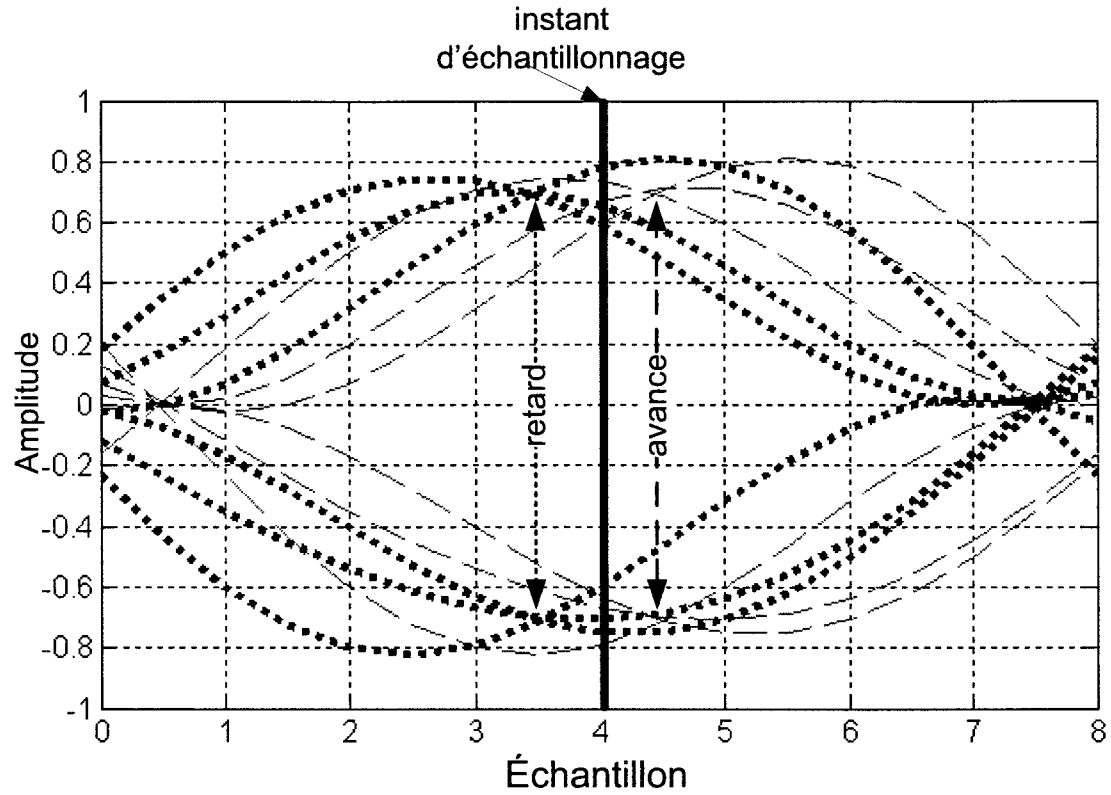


FIGURE 3.23 Effet d'un décalage d'un demi échantillon

constitué de lignes entrecoupées est le résultat de l'échantillonnage en avance par rapport au bon instant. Leurs instants de plus grande ouverture sont aux échantillons 3.5 et 4.5 respectivement. Le système considère que le symbole doit être lu au quatrième échantillon puisqu'il n'est pas en mesure de faire une interpolation pour corriger ces erreurs de demi-échantillon.

### 3.8 Simulations et résultats

Nous avons simulé le système présenté sur Simulink pour des valeurs de SNR allant de 8 dB à 13 dB avec des incréments de 0.1 dB. Les paramètres utilisés sont indiqués au tableau 3.2. Nous avons fixé le temps de simulation à un maximum de  $10^7 T_B$ , où  $T_B$  est la période d'un bit de donnée. Cependant, afin de limiter la durée des simulations, nous

Élément bloc	Paramètre	valeur
Format de paquet	Nombre de séquences de Barker	4
	Nombre de séquences PRBS	10
Filtre de Nyquist	Longueur	65
	$\beta$	0.22
Détection de début de paquet	Seuil	6
Filtre d'égalisation	Nombre de taps	1
	$\mu$	0.025
AGC	$\alpha_{AGC}$	0.9
	Référence (dB)	0
PLL	$N_{DDFS}$	128
	$\alpha_{PLL}$	0.99
	$K_{PLL}$	1
Recouvrement d'horloge	$\alpha_{CLK}$	0.99

TABLE 3.2 Liste des paramètres du système utilisés lors de la simulation

avons limité le nombre d'erreurs binaires à 1000 erreurs.

La figure 3.24 illustre les résultats obtenus. La courbe pointillée est la limite théorique établie par référence au tableau 1.2 pour le QPSK. Nous pouvons observer que :

- La tendance de la courbe des résultats de simulation suit celle de la courbe théorique. Il y a une décroissance exponentielle du BER lorsque le SNR augmente.
- Il existe une perte de l'ordre d'au plus 0.3 dB entre la courbe théorique et la courbe de simulation. Ainsi, pour atteindre le taux d'erreurs binaires mentionné par la courbe théorique, nous devons transmettre avec une puissance de 0.3 dB supplémentaire. Cette différence provient de plusieurs sources d'imperfections du système. Par exemple, au niveau du PLL, nous utilisons seulement une résolution de 128 phases. Ceci équivaut à avoir une résolution de  $\pi/64$  au niveau de la détection de phase et de fréquence.

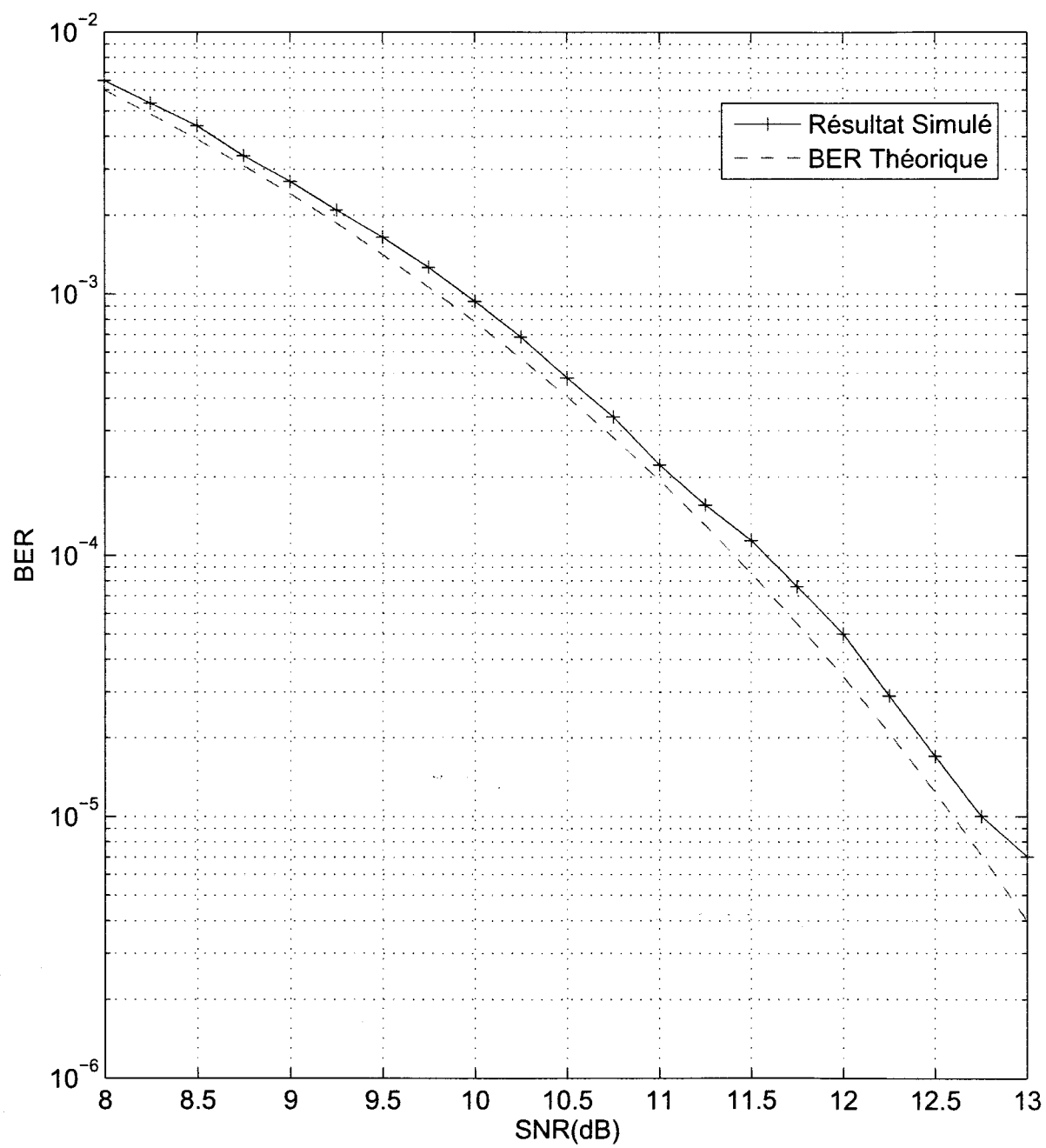


FIGURE 3.24 Résultat de simulation sur Simulink



## CHAPITRE 4

### LES APPAREILS DE PROTOTYPAGE

Dans ce chapitre, nous faisons un survol sur les outils matériels et logiciels qui ont été utilisés dans le cadre de ce projet. Nous présenterons d'abord les plateformes de développement, notamment leurs architectures et composantes. Ensuite, nous ferons un bref survol des outils logiciels.

#### 4.1 Vue d'ensemble du matériel

La plateforme utilisée pour notre réalisation est constituée d'une boîte dans laquelle se trouvent deux cartes de développement dont un Signal Master Quad (SM-QUAD) et un VHS-ADC (contenant un module DAC Daughter board) présentés à la figure 4.1. Les cartes y sont connectées à travers un bus PCI. Quatre prises Ethernet sont aussi disponibles pour permettre la connexion des appareils au réseau local. Comme nous le verrons à la section 4.4, il est possible de programmer les cartes à travers une connexion Ethernet.

Par ailleurs, il existe plusieurs techniques de transmission entre les cartes. Lyrtech Inc. a développé un connecteur LYRIO+ [Lyrtech DRC V4] qui utilise une connexion RapidCHANNEL pour permettre une communication rapide entre le SMQUAD et le VHS-ADC en passant par un module Dual RapidChannel (DRC). Cette connexion permet de faire le transfert de données avec une largeur de bande de 1GBps. Dans ce système, elle est la plus performante en termes de taux de transfert de données pour les communications entre les cartes. Il existe aussi d'autres types de liens de communications tel que le RTDex mais nous ne les décrirons pas car ils ne sont pas utilisés dans le cadre de ce projet.

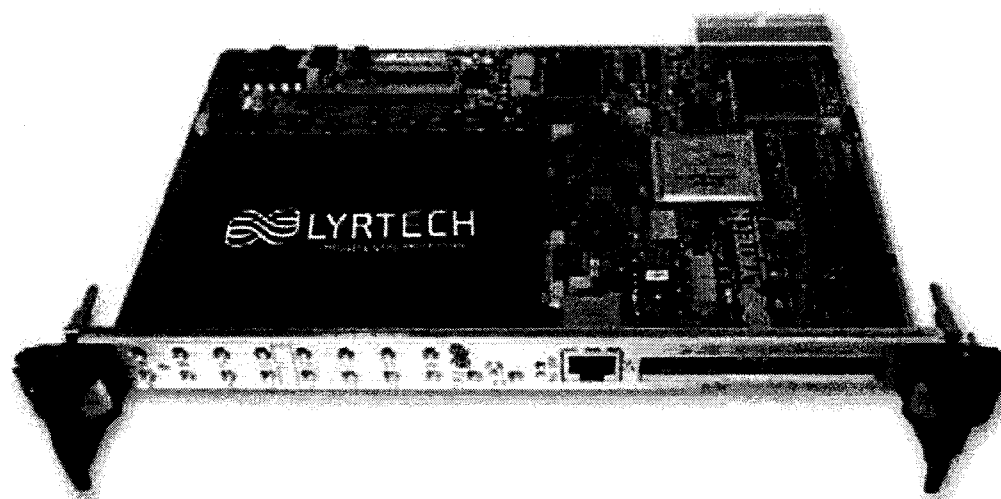
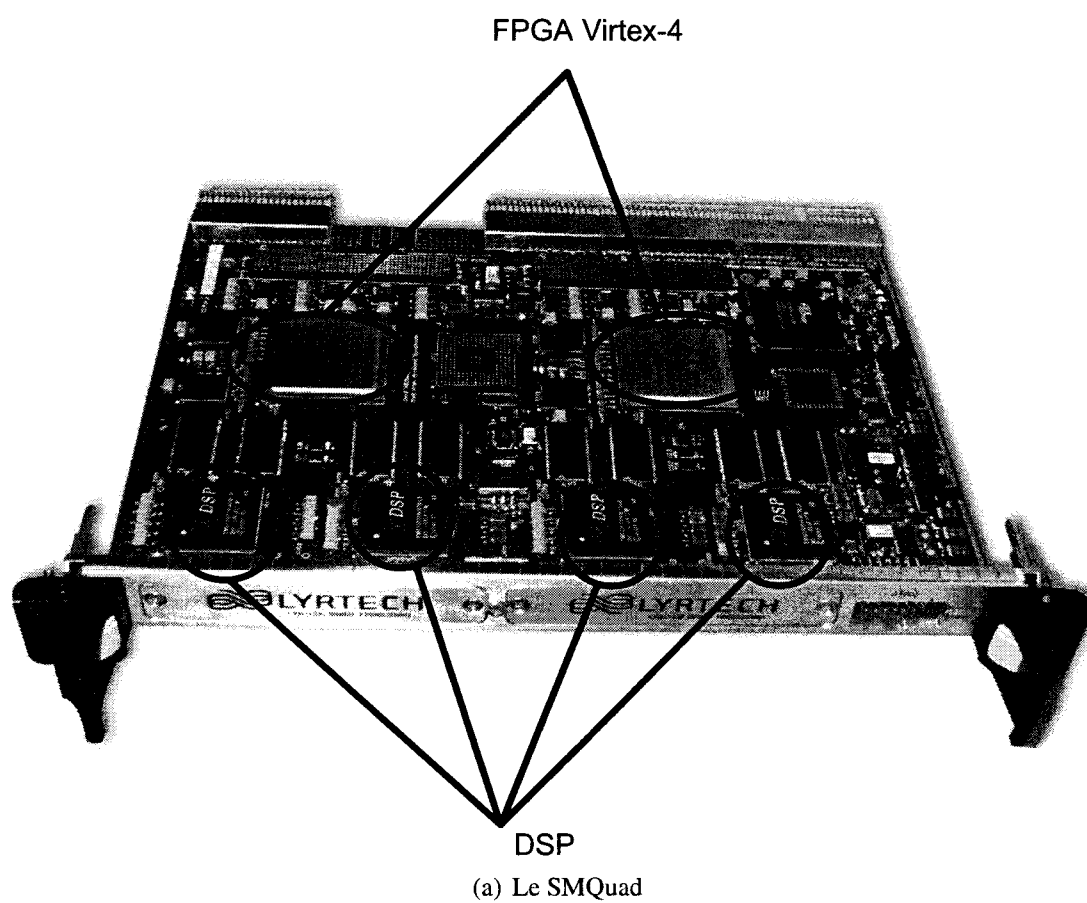


FIGURE 4.1 Photos des plateformes de développement

## 4.2 Description hardware du Signal Master QUAD

Le Signal Master QUAD contient deux FPGA Virtex-4 xc4vlx100 de Xilinx Inc. et quatre DSP C6416 1GHz de Texas Instruments qui sont tous programmables par l'utilisateur. Un module supplémentaire peut y être intégré afin d'ajouter des fonctionnalités supplémentaires. Sur notre plateforme, un module DRC est présent pour permettre une communication RapidCHANNEL. La figure 4.2 montre un plan simplifié de l'agencement des principales composantes que nous avons utilisées et de leur interconnexions.

### FastBus

Deux DSPs sont connectés à chacun des FPGAs. Le Fastbus est l'interface principale que nous utilisons pour le transfert de données. Il utilise un EMIF de 32 bits en mode synchrone pour permettre, comme son nom l'indique, une communication rapide multi-canaux [Lyrtech FastBus].

Sur chaque DSP, il existe 8 canaux de communications à travers l'interface FastBus pour effectuer de la transmission et de la réception, soit un total de 16 canaux.

La figure 4.3(a) montre l'interface de configuration Simulink du bloc FastBus du côté DSP ainsi que les paramètres qui y sont associés tel que fourni par Lyrtech Inc. À noter les paramètres suivants :

*Channel ID* - Ceci se réfère au canal que nous utilisons pour la communication avec le FPGA. Pour chaque DSP et chaque mode de transfert (transmission ou réception), il y a 8 canaux utilisables.

*Channel Direction* - Il détermine si le bloc du côté DSP est en réception de données du FPGA ou en transmission de données vers le FPGA.

*Transfer mechanism* - Deux types de transfert sont possibles : le Event-triggered DMA et le Software-triggered DMA. Lors de nos modèles, nous utilisons le Event-triggered

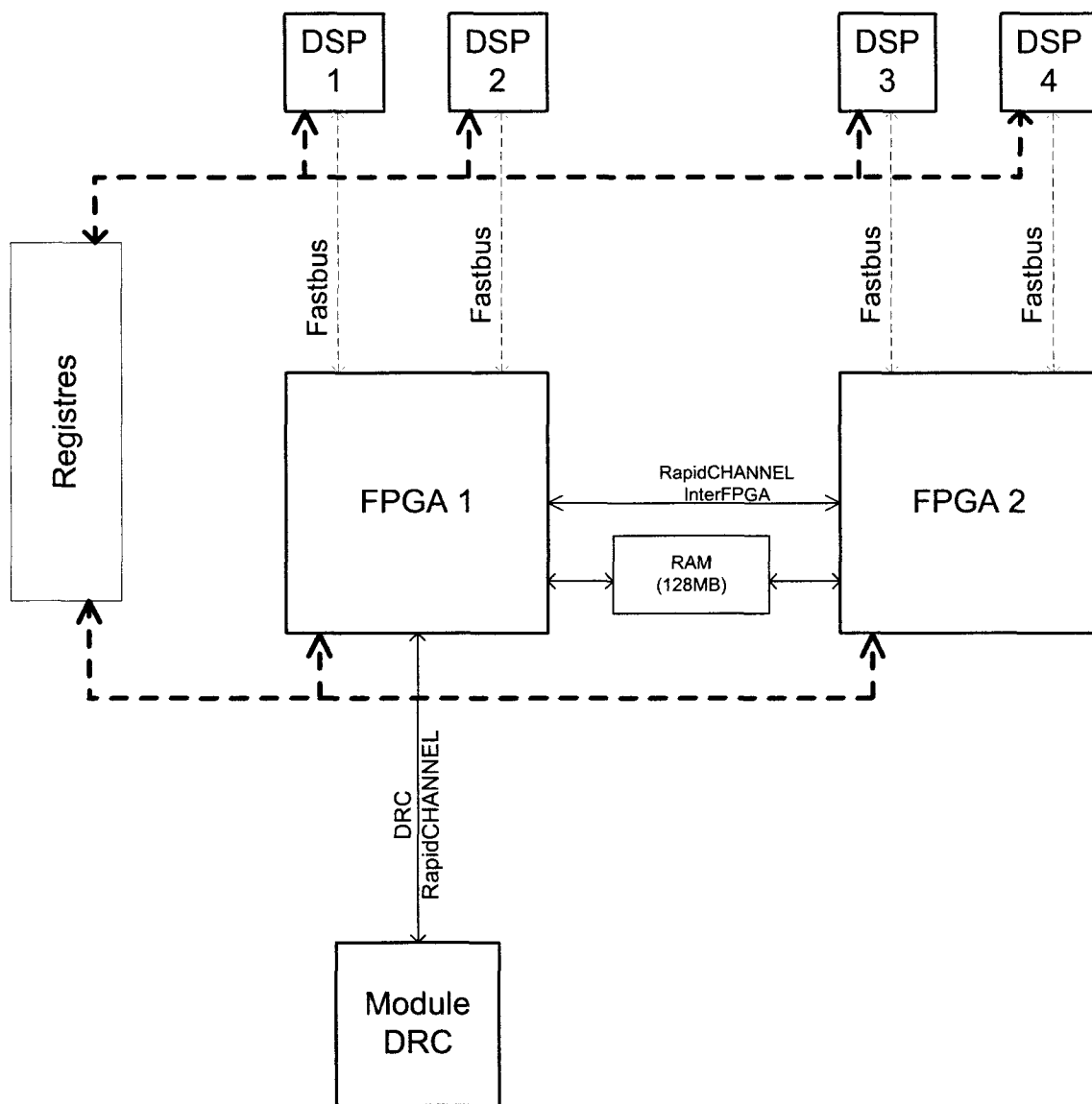
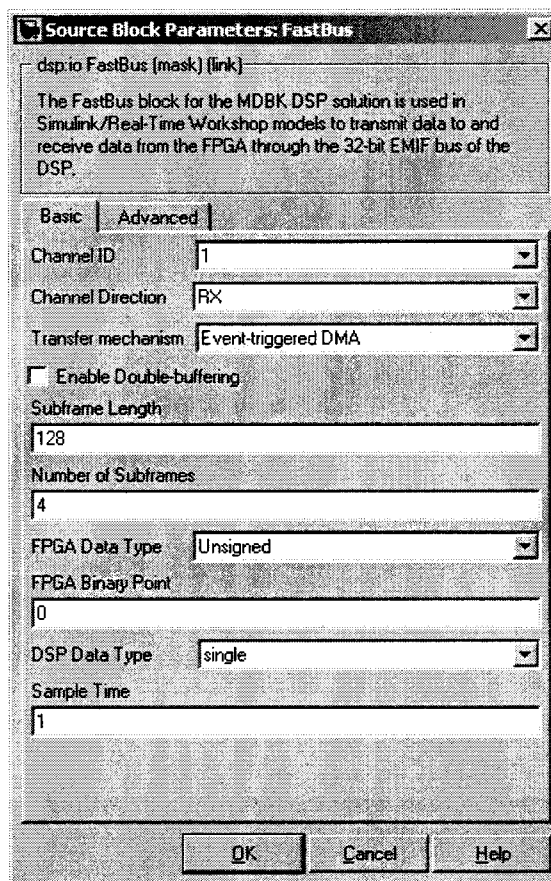
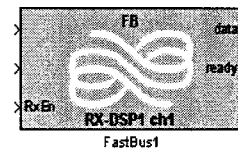
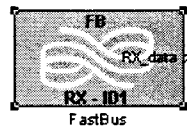
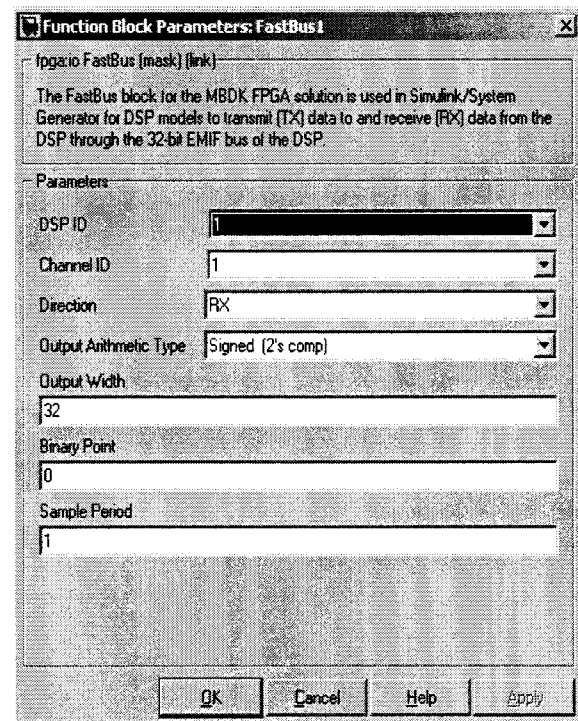


FIGURE 4.2 Composantes du SMQUAD et leurs interconnexions



(a) Le bloc côté DSP



(b) Le bloc côté FPGA

FIGURE 4.3 Les blocs FastBus

DMA car il offre une meilleure performance au niveau de la latence et du nombre de cycles du CPU utilisés [Lyrtech FastBus].

*Subframe Length et Number of Subframes* - Le produit de ces paramètres détermine la dimension de chaque trame de données à être traitée par le DSP. La longueur de chaque sous-trame est limitée à 1023.

*FPGA Data Type, FPGA Binary Point et DSP Data Type* - Comme leurs noms l'indiquent, ils définissent le type de données et le point binaire vus par le bloc FPGA ainsi que le type de données utilisé sur le DSP. En posant les valeurs appropriées à chacun de ces paramètres, nous effectuons la conversion d'un type de représentation de données à un autre.

À la figure 4.3(b), le bloc FastBus du côté FPGA et les paramètres associés sont illustrés. Les points importants à noter sont les suivants :

*Port Ready* - Lorsque le FastBus est prêt à transmettre/recevoir des données, il émet un drapeau 1.

*Port TxEn ou RxEn* - Une entrée à actif correspond à une demande du FPGA de transmettre ou recevoir des données.

*DSP ID* - Les valeurs possibles sont 1 ou 2. Lorsque ce bloc se trouve sur le FPGA, elles correspondent aux DSP 1 et 2 respectivement. Lorsque ce bloc se trouve sur le FPGA 2, elles correspondent aux DSP 3 et 4 respectivement

*Direction* - La direction se fait ici par rapport au FPGA. Ainsi, RX signifie que le FPGA est récepteur de données alors que TX signifie que le FPGA est transmetteur de données.

Lyrtech Inc. mentionne que le pilote du FastBus peut opérer en mode bloquant (synchrone) ou non-bloquant (asynchrone) [Lyrtech FastBus]. En conséquence, il pourrait arriver à la suite d'un design en mode bloquant que les DSPs arrêtent de fonctionner à une itération. Afin d'éviter de telles situations, l'utilisation de la priorité des blocs est recommandée [Lyrtech VHS UserGuide].

## **RapidChannel**

Les communications RapidChannel permettent des transferts à haut débit (1GBps) entre deux composantes FPGA. Elles utilisent 8 canaux de 1 octet avec une horloge fonctionnant à 125 MHz pour atteindre ce débit [Lyrtech DRC V4]. Lors de nos designs, nous l'utilisons pour faire une connexion entre les deux composantes FPGA du SMQUAD ou pour une connexion entre un des deux FPGA et le module DRC connecté au VHS-ADC. L'interface de configuration du bloc RapidChannel est illustrée à la figure 4.4. Les principaux paramètres sont les suivants :

*Platform Type* - Les trois choix possibles pour le type de plateforme sont le SMQUAD, le module DRC et le VHS-ADC/DAC.

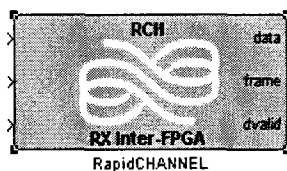
*Interface* - L'interface Inter-FPGA correspond à une communication RapidChannel entre deux FPGA situés sur la même plateforme (les deux FPGA du SMQUAD). Nous utilisons l'interface LYRIO pour effectuer la communication avec une autre carte qui est en général le VHS-ADC.

## **Registres**

Le SMQUAD possède 16 registres de contrôle (custom registers) qui peuvent être accédés en lecture et en écriture par les composantes DSP et FPGA [Lyrtech VHS UserGuide]. Comme la vitesse de communication entre les composantes DSP et FPGA à travers ces registres est limité, ces registres sont principalement utilisés à des fins de contrôle.

## **Enregistrement de données vers la mémoire RAM**

Un total de 128 MB de mémoire RAM est accessible pour chaque FPGA du SMQuad. Le bloc *SDRAM Record* du FPGA est utilisé pour l'écriture de grosses quantités de données



**Function Block Parameters: RapidCHANNEL**

fgpio RapidCHANNEL (mask) (link)

The RapidCHANNEL block is used in Simulink/System Generator for DSP models to send and receive data between FPGA through RapidCHANNEL interfaces.

**Parameters**

Platform Type: SMDUAD

Interface: Inter-FPGA

Direction: RX

Output Arithmetic Type: Unsigned

Output Width: 64

Binary Point: 0

Sample Period: 1

☐ Provide Control Signals

☐ Provide Programmable I/O

OK Cancel Help Apply

FIGURE 4.4 Interface Simulink du bloc RapidChannel.



vers la mémoire SDRAM du FPGA qui sont éventuellement sauvegardées dans un fichier binaire spécifié par l'utilisateur à travers l'interface usager de l'utilitaire décrit à la section 4.4. Il existe un bloc connexe (SDRAM Playback) qui peut être utilisé pour la lecture de données de la mémoire. L'enregistrement ou la lecture de données requièrent chacun un bloc de contrôle (les blocs *FPGA SDRAM Record Control* ou *FPGA SDRAM Playback Control* respectivement) sur le DSP. Ce dernier offre alors la possibilité à un utilisateur de déclencher l'enregistrement ou la lecture de données à travers une interface. Ceci est davantage détaillé à la section 4.4.

Du côté DSP, nous notons les points intéressants suivants :

*Recording Source* - Deux sources sont disponibles, notamment une rampe générée à l'interne sur le FPGA (pour des fins de vérifications du fonctionnement) et les signaux de réception aux ports du bloc.

*Trigger Mode* - Ce paramètre indique si le bloc doit effectuer l'enregistrement d'une trame de données (*One Shot*) ou enregistrer des séquences de trames jusqu'à nouvel ordre (*Continuous*).

*Trigger Source* - Il permet de décider si le début de l'enregistrement se fait manuellement (*Manual*), à partir d'un des canaux d'enregistrement (*Signal*) ou selon un flag indiqué dans le modèle FPGA (*External FPGA Trigger*).

Du côté FPGA, les points intéressants sont les suivants :

*Port Trigger* - Ce port est utilisé pour signaler au bloc de débiter l'enregistrement des données. Elle n'apparaît que lorsque la case *Provide Trigger Signals* est cochée.

*Port Done* - Comme son nom l'indique, ce port indique si le bloc a terminé l'enregistrement des données.

*Port valid* Ce port permet de sélectionner les données qui sont valides.



FPGA SDRAM  
Record control

**Block Parameters: FPGA SDRAM Record control**

dsp.control FPGA SDRAM Record control (mask) (link)

Use the FPGA SDRAM Record control block to record large amounts of data in the SDRAM of the FPGA for offline processing with MATLAB or other software.

Parameters

Recording Source: **FPGA SDRAM Record**

Number of Channels: **1**

Trigger Mode: **One-Shot**

Trigger Source: **Signal**

Trigger Select: **1**

Trigger Level: **0**

Trigger Polarity: **Rising**

☐ Arm

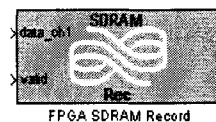
Frame Size: **1024\*1024**

Recording Offset: **0**

Sample Time: **1**

OK Cancel Help Apply

(a) Bloc de contrôle du DSP



FPGA SDRAM Record

**Function Block Parameters: FPGA SDRAM Record**

fpga.io FPGA SDRAM Record (mask) (link)

Use the FPGA SDRAM Record block for the MBDK FPGA solution with a Simulink/System Generator for DSP FPGA model to write large amounts of contiguous data to the onboard SDRAM of the FPGA.

Parameters

Number of Channels: **1**

Output Arithmetic Type: **Signed (2's comp)**

Binary Point: **0**

Sample Period: **1**

☐ Provide Trigger Signals

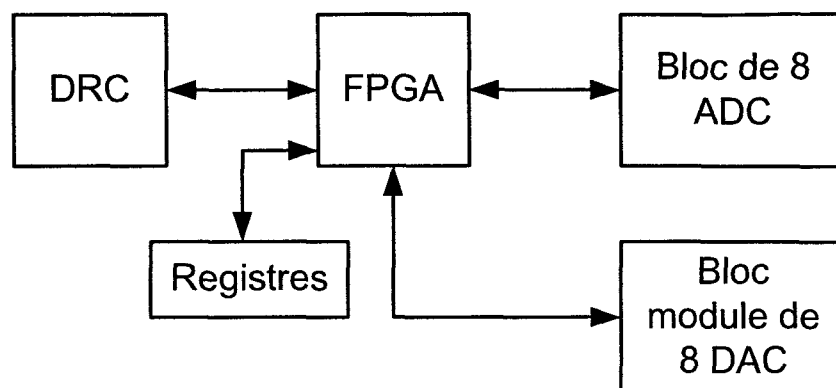


FIGURE 4.6 Diagramme bloc des principales composantes du VHS-ADC

### 4.3 Description hardware du VHS-ADC

Le VHS-ADC possède un FPGA pouvant être programmé selon les besoins de l'utilisateur. Outre les 8 convertisseurs analogiques-numériques (ADC) présents, un module de 8 convertisseurs numériques-analogiques (DAC) est présent. De plus, il est possible de faire une communication avec le SM-Quad à travers un RapidChannel. Le diagramme bloc des principales composantes de cette carte est illustré à la figure 4.6.

Le bloc *FPGA Board Configuration* est utilisé pour définir les paramètres du FPGA à utiliser, notamment le type de plateforme (VHS-ADC ou VHS-DAC) pour lequel sera compilé le modèle. Le FPGA qui se trouve sur la carte est un Virtex-4 xc4vsx55 de Xilinx Inc.

### ADC et DAC

Les convertisseurs analogique-numérique ont une fréquence d'échantillonnage fixée à 104 MHz. Il est aussi possible de changer cette fréquence d'horloge à 52 MHz à travers le bloc ADC Control du DSP ou d'utiliser une source externe. Les convertisseurs numériques-analogiques se trouvent sur un module complémentaire fourni par Lyrtech

Inc. Ces convertisseurs échantillonnent les données à un taux dicté par le FPGA du VHS-ADC. Ils héritent donc de l'horloge des ADC. Les gains d'amplification de ces modules sont fixés. L'interface de configuration de ces blocs est illustrée à la figure 4.7.

Pour changer le nombre de convertisseurs utilisés, il suffit de choisir le nombre désiré dans la liste déroulante du paramètre *Number of channels*.

### **RapidChannel**

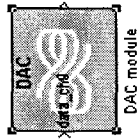
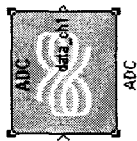
Comme nous l'avons mentionné, le FPGA du VHS-ADC peut communiquer avec des FPGA situés sur d'autres plateformes à travers le RapidChannel. Comme ce bloc est identique à celui décrit à la section 4.2, nous ne le décrivons pas de nouveau.

### **Registres**

Un total de 12 registres de contrôle (Custom Registers) se trouvant sur le VHS-ADC peuvent être contrôlés par les composantes DSP. Les registres sont alors identifiés sur le DSP selon la plateforme sur laquelle ils se trouvent (voir figure 4.8(b), paramètre *Board ID*).

## **4.4 Description software**

Dans cette section, nous faisons une description générale des composantes logicielles reliées à l'utilisation des plateformes.



**Function Block Parameters: ADC**

fga to ADC module (mask) [link]

The ADC block is used with a Simulink/System Generator for DSP model to access the digital data of the VHS-ADC analog-to-digital converters.

Note: To use this block, an ADC module must be connected to your system.

Parameters

Number of channels: [1]

Binary point: [0]

Sample period: [1]

☐ Provide overrange flags

☐ Provide trigger signals

OK Cancel Help Apply

(a) Convertisseur analogique-numériques

**Function Block Parameters: DAC module**

fga to DAC module (mask) [link]

The DAC module block is used with a Simulink/System Generator for DSP model to access the digital data of the DAC module's digital-to-analog converters.

Note: To use this block, an DAC module must be connected to your system.

Parameters

Number of Channels: [1]

Binary Point: [0]

Sample Period: [1]

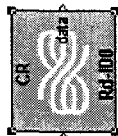
☐ Provide Trigger Signals

☐ Provide Run Signal

OK Cancel Help Apply

(b) Convertisseur numériques-analogique

FIGURE 4.7 Les convertisseurs et leurs paramètres



**Function Block Parameters: Custom register**

The Custom register block allows you to access 12 memory-mapped registers in reading or writing.

Parameters

Register ID: 0

Direction: Read

Output arithmetic type: Unsigned

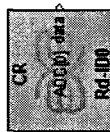
Output width: 32

Binary point: 0

Sample period: 1

Buttons: OK, Cancel, Help, Apply

(a) Sur le FPGA



**Source Block Parameters: VHS-ADC/DAC Virtex-4 C...**

dsp control VHS-ADC/DAC Virtex-4 custom register [mask] [link]

The VHS-ADC/DAC Virtex-4 custom register block for the MBDK DSP solution is used in Simulink/Real Time Workshop models to remotely define custom values for the instantiated custom register of the FPGA design through SMCCE.

Parameters

Carrier type: VHS-ADC Virtex-4

Board ID: Board [0]

Register ID: 0

Direction: Read

FPGA data type: Signed(2 comp)

FPGA binary point: 0

DSP data type: single

Sample time: 1

Buttons: OK, Cancel, Help

(b) Sur le DSP

FIGURE 4.8 Blocs des Registres du VHS-ADC

## Le SMCCE

Le Signal Master Central Communication Engine (SMCCE) est un logiciel qui agit comme un pont entre l'ordinateur hôte sur lequel se trouvent les plateformes et la station de travail. Comme nous le constatons à la figure 4.9, les plateformes sont connectées à un ordinateur à travers un bus PCI. Pour démarrer le SMCCE, l'utilisateur peut pointer vers le raccourci *Start\Lyrtech\Host SDK\cPCI Platform\SMCCE*. Ceci permettra alors aux utilitaires de détecter les plateformes de prototypage rapide de Lyrtech Inc. disponibles pour les programmer et effectuer des tests. Le SMCCE a aussi comme tâche de s'occuper du transfert d'information telles que les requêtes de la station de travail (e.g. programmation des DSP et exécution). Finalement, il coordonne l'échange de données (par exemple les données enregistrées dans la mémoire de la carte) entre la station de travail et la plateforme. Lors de l'implémentation, il est nécessaire de s'assurer que le SMCCE est en état fonctionnel pour garantir une bonne communication avec la plateforme désirée.

## Enregistrement de données

L'enregistrement des données à l'aide de la mémoire disponible dans le FPGA est très utile lors du design pour la vérification du bon transfert des données. À cet effet, la composante logicielle *Record and Playback Utility* permet de récolter les données et de les écrire dans un fichier binaire pour faire du traitement hors ligne.

Pour pouvoir enregistrer les données, un bloc *FPGA SDRAM Record* ainsi qu'un bloc *FPGA SDRAM Record control* doivent être présents sur le FPGA et un DSP respectivement. L'utilisateur indique à l'utilitaire le DSP qui contrôle l'enregistrement des données en cliquant sur le bouton *Lyrtech platform selection* (voir figure 4.10). Au champ *Filename*, l'utilisateur décide du fichier binaire vers lequel s'effectue l'enregistrement des données. Pour mettre l'utilitaire en mode d'écoute, il suffit d'appuyer sur le bouton *Get Next Available Frame*. Le bouton *Stop After Next Acquisition* apparaîtra alors pour que l'utilisateur puisse

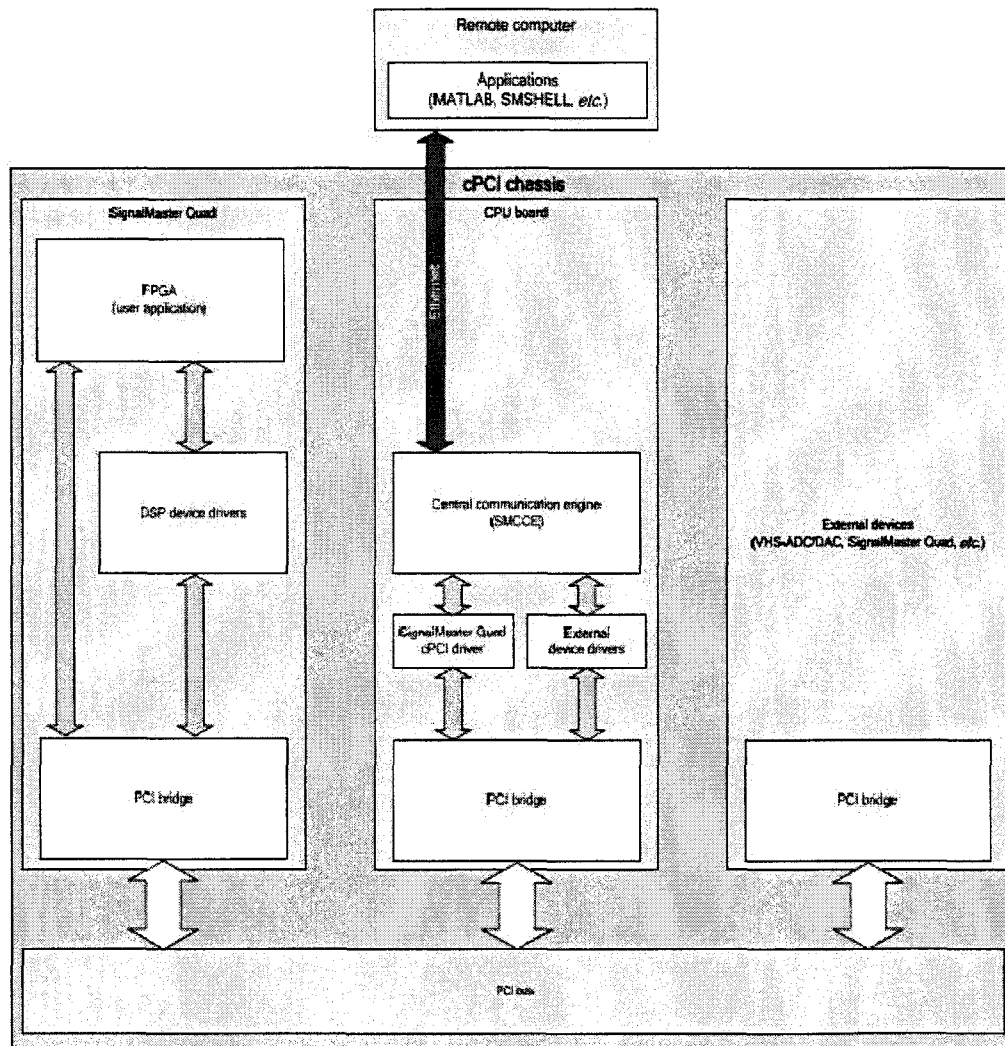


FIGURE 4.9 Le SMCCE comme un pont de communications



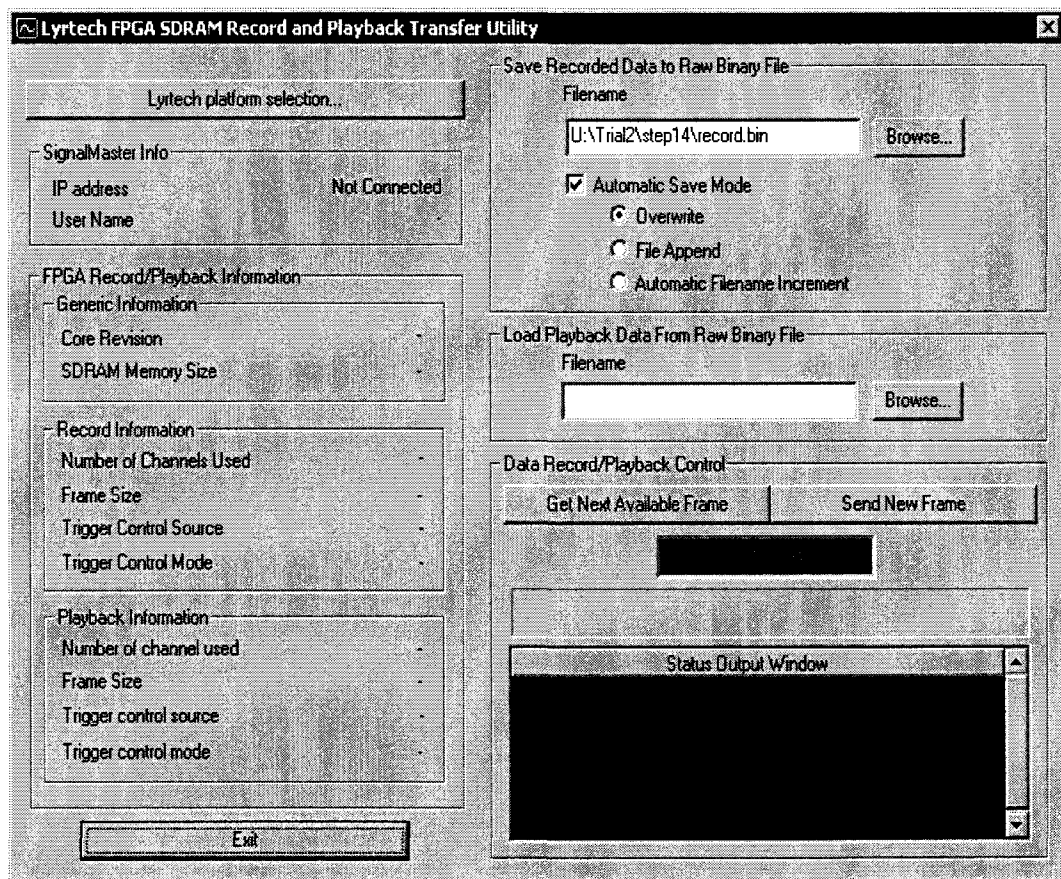


FIGURE 4.10 Interface Graphique du Record and Playback Utility

décider si l'acquisition des données doit s'arrêter après la fin de la prochaine trame.

Le contrôle de l'enregistrement des données s'effectue à partir du bloc situé au niveau du DSP. Par exemple, lorsque le *Trigger Source* est à *Manual*, l'enregistrement commence à partir du moment où l'utilisateur coche la case *Trigger*. Le nombre de données valides enregistrées  $N_{record}$  est déterminé par le paramètre *Frame Size*. La validité est approuvée par le port *valid* sur le bloc du DSP (voir la figure 4.5).

Pour récupérer les données, nous pouvons utiliser le circuit présenté à la figure 4.11(a). À la figure 4.11(b), le paramètre *Packing Mode* est à 2 car les données sont enregistrées de cette façon [Lyrtech SDRAM].

#### 4.5 Outils nécessaires au développement

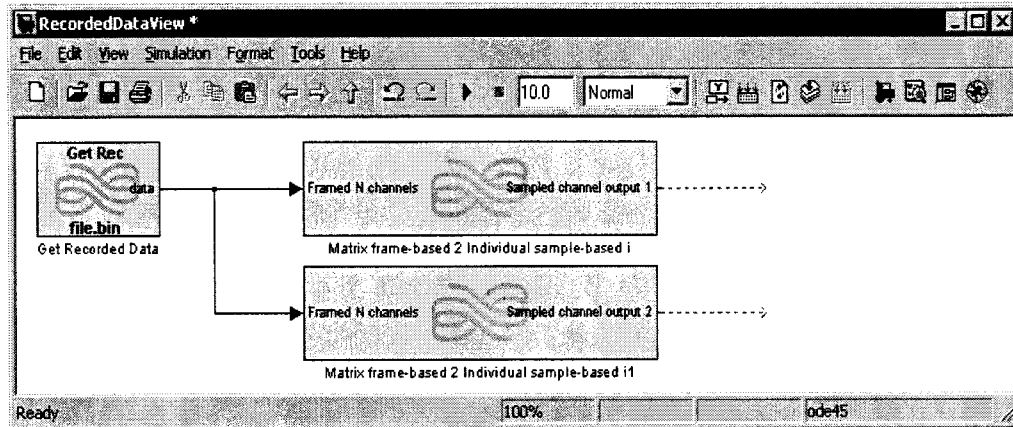
Lyrtech Inc. propose deux types de développement pour la plateforme. Le Board Software Development Kit (BSDK) est principalement la librairie fournie pour faire de la programmation en langage C dans le cas du DSP ou en VHDL dans le cas du FPGA. Nous avons opté pour utiliser l'autre approche basée sur le Model-Based Development Kit (MBDK) qui est la librairie fournie pour faire du développement en utilisant des blocs de Simulink.

La figure 4.12 illustre les étapes successives lors du développement d'un prototype. Chaque modèle DSP et FPGA sont développés séparément.

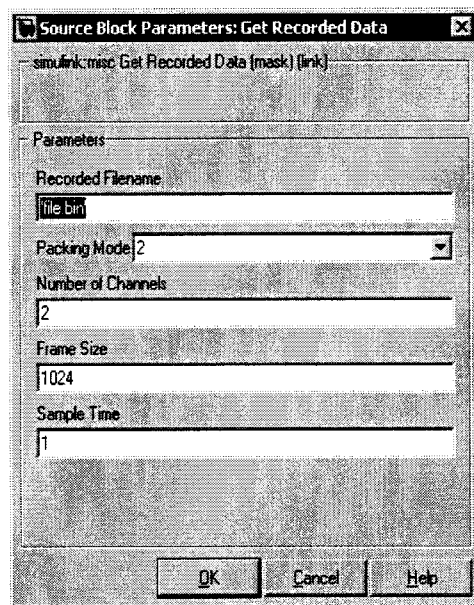
Pour la création de modèles DSP, les outils nécessaires sont :

- Simulink
- Real-Time Workshop
- Code Composer Studio (Version 3.0)

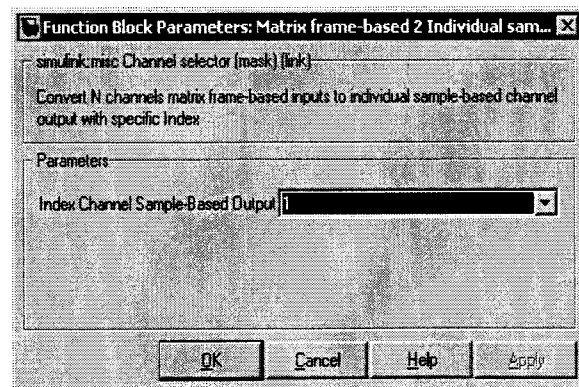
Pour chaque DSP, un modèle Simulink doit être créé. À la compilation de chacun d'entre eux, Real-Time Workshop génère du code C (les fichiers .c et .h) en se basant sur les blocs utilisés et les fonctionnalités demandées. Les outils de Code Composer Studio compilent



(a) Circuit pour la récupération des données d'un fichier binaire



(b) Paramètres du bloc de récupération des données



(c) Paramètre du bloc de choix de canal

FIGURE 4.11 Lecture des données enregistrées à partir d'un fichier binaire

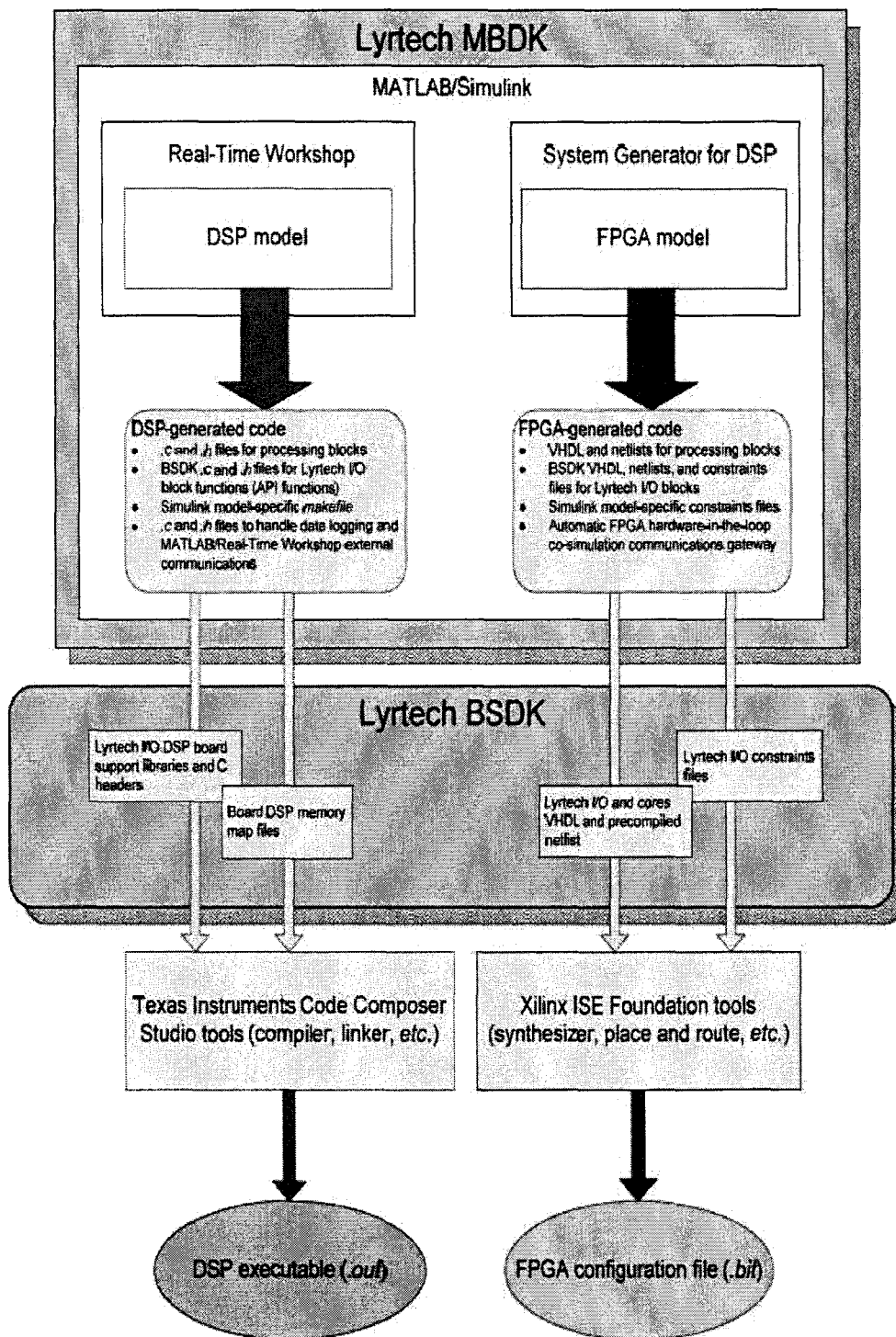


FIGURE 4.12 Étapes de développement [Lyrtech SMQ DesignGuide].

le code et créent les liens (build) pour produire l'exécutable en format *fichierDSP.out*.

Pour la création de modèles FPGA, les outils nécessaires sont :

- Simulink
- Xilinx System Generator
- Xilinx ISE Foundation tools

Pour chaque FPGA, un modèle Simulink doit être créé. À la compilation, Xilinx System Generator génère les fichiers VHDL et les netlists nécessaires. Les outils de Xilinx Foundation Tools effectuent les étapes finales (synthesizer, xflow, etc) pour produire l'exécutable (bitstream) en format *fichierFPGA.bit*.

Les étapes à suivre pour la création d'un modèle sont davantage détaillées au chapitre 5.

## CHAPITRE 5

### RÉALISATION ET PERFORMANCE

Ce chapitre est divisé en trois sections. Dans la première section, nous effectuons une description détaillée du transmetteur étape par étape. La deuxième section explique les grandes lignes de la démarche utilisée pour réaliser le récepteur. Finalement, nous décrivons les performances mesurées avec cette réalisation.

Au courant de la réalisation, nous avons utilisé le complément à 2 pour la représentation des données signées.

#### 5.1 Le transmetteur

Pour réaliser le transmetteur, nous avons d'abord utilisé un DSP pour agir comme source des paquets à transmettre. Sur le FPGA du SMQuad, nous effectuons par la suite un suréchantillonnage des données. Une connexion RapidChannel est utilisée pour transférer les données vers la carte VHS-ADC. Nous y effectuons la mise en forme d'impulsions par un filtre en racine carrée cosinus surélevé avant la transmission vers les convertisseurs numériques-analogiques. Le schéma bloc du transmetteur est illustré à la figure 5.1. Nous avons subdivisé l'implémentation du transmetteur en quatre étapes.

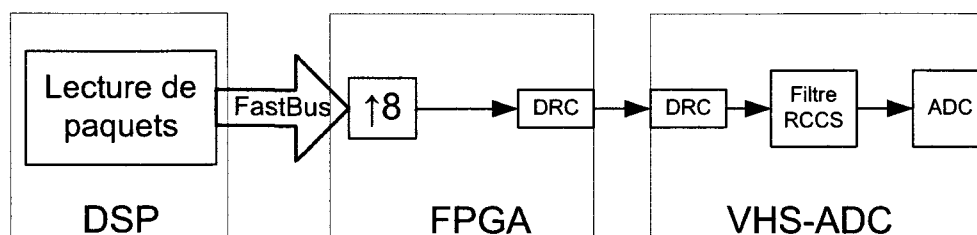


FIGURE 5.1 Diagramme bloc du transmetteur

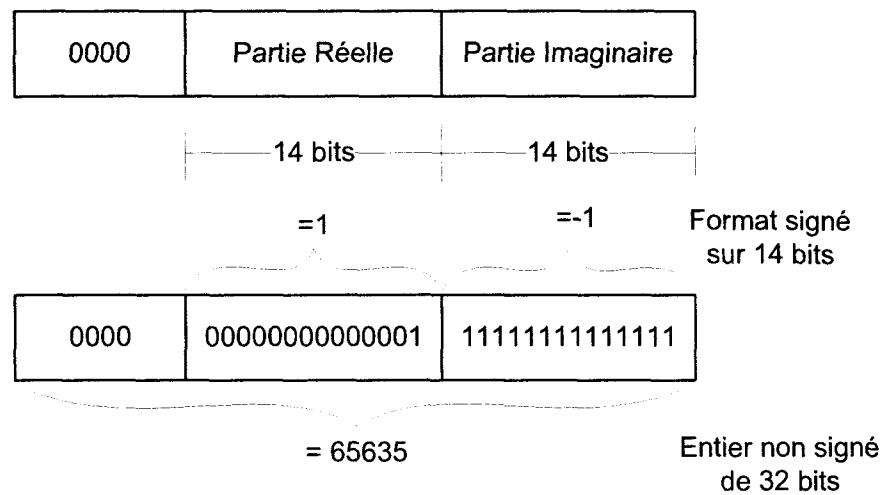


FIGURE 5.2 Représentation et interprétation de chaque symbole

**Étape 1 : Formation du paquet**

Chaque paquet transmis a la structure décrite à la section 3.1. Quatre séquences de Barker de longueur 7 sont utilisées pour la détection du paquet. Les séquences de synchronisation ont été formées à partir d'un ensemble de 9 séquences  $S$  et d'une séquence  $S$  inversée. La séquence binaire de données est générée à partir de quatre registres à décalages tous initialisés à 1. La boucle de rétroaction utilisée est  $x_3 \oplus x_4$ . Cette séquence est répétée et modulée en QPSK de sorte à avoir des paquets d'une longueur totale de 512 symboles. Donc, par paquet, nous transmettons 183 symboles dans l'entête et 329 symboles de données.

Les convertisseurs numérique-analogique utilisent une représentation sur 14 bits en format complément à 2. Nous avons donc représenté les parties réelles et imaginaires de chaque symbole selon ce format. Quatre zéros sont ajoutés au début pour former une donnée de type entier non-signé de 32 bits. La figure 5.2 montre une représentation pour chaque symbole tel que généré et la façon de l'interpréter.

La compagnie Lyrtech Inc. fournit le bloc *bit packer* qui permet de concaténer des bits de plusieurs sources pour en faire une variable de type entier non-signé de 32 bits. Ainsi,

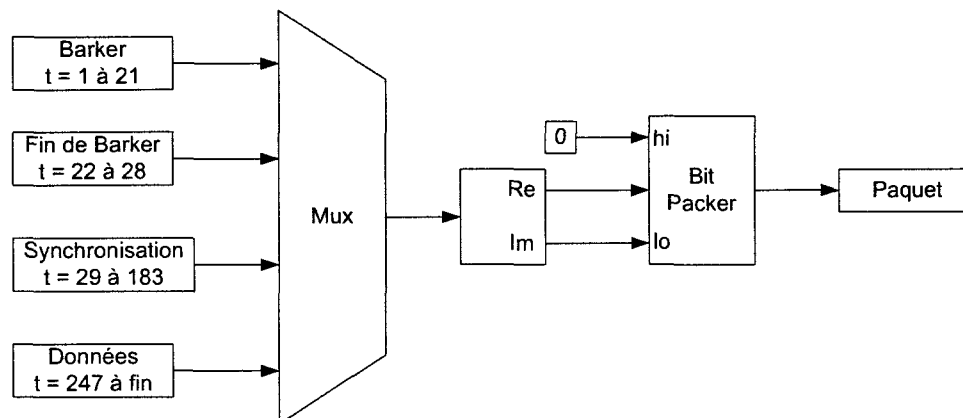


FIGURE 5.3 Circuit utilisé pour générer les paquets

comme illustré à la figure 5.3, ce bloc est utilisé pour générer le paquet de transmission. Nous spécifions au bloc que la première entrée forme les 4 premiers bits. Les deux entrées suivantes doivent être représentées sur 14 bits et elles sont de type signé. Ceci est fait en spécifiant le vecteur  $[4 - 14 - 14]$  au champs *[Input Descriptor]* du bloc. Le signe négatif indique que l'entrée est signée. Les paquets de transmission sont donc préalablement générés par l'exécution d'un modèle Simulink.

## Étape 2 : Design du DSP

Le DSP est utilisé à des fins de contrôle. Il permet de définir l'horloge utilisée pour différentes composantes du système. Les détails du design du DSP sont décrits à l'annexe II.1.

## Étape 3 : Design du FPGA(SMQuad)

Comme nous l'avons mentionné, le FPGA du SMQuad a pour tâche d'effectuer un sur-échantillonnage des données reçues par le FastBus du DSP et de les acheminer vers le VHS-ADC à travers le RapidCHANNEL. Les détails du design de ce FPGA est fourni à



l'annexe II.2

#### **Étape 4 : Design du FPGA(VHS)**

Au niveau de la composante FPGA du VHS-ADC, nous effectuons la mise en forme d'impulsions à travers un filtre en racine carré cosinus surélevé. Les données sont transmises vers un module convertisseur numérique-analogique qui transmet le signal sur deux canaux pour les composantes en phase I et en quadrature Q. Les détails du design de ce FPGA est fourni à l'annexe II.3

Suite à la génération du bitstream, ce FPGA est programmé avant de procéder à l'étape suivante.

#### **Connexion au DSP et exécution**

Dans un premier temps, nous devons indiquer que la composante FPGA du SMQuad doit être programmée avec le bitstream *SMQUADFPGA1.bit*. À cet effet, le bloc *Set bitstream* que nous avons inséré sur le modèle du DSP s'avère utile. En double-cliquant sur ce bloc, une fenêtre apparaîtra pour que l'utilisateur fasse la sélection du bitstream approprié. Alternativement, l'utilisateur peut effectuer cette commande en allant dans le *Configuration Parameters\Real-Time Workshop\Interface* et ajouter la commande '*fpgaload*' '*SMQUADFPGA1.bit*' à côté du '*autodetect*'.

Pour se connecter au DSP, il suffit de choisir *Connect to target* dans le menu *Simulation* ou d'utiliser le bouton disponible dans la barre à outils (voir figure II.1). Une fenêtre apparaîtra alors demandant à l'utilisateur de choisir la plateforme (voir figure 5.4). En appuyant sur le bouton *Connect to Lyrtech development platform*, le FPGA sera programmé avec le bitstream et l'exécutable *.out* sera programmé sur le DSP correspondant. L'exécution du code débute lorsque l'utilisateur appuie sur le bouton *play* situé à côté du *connect*.

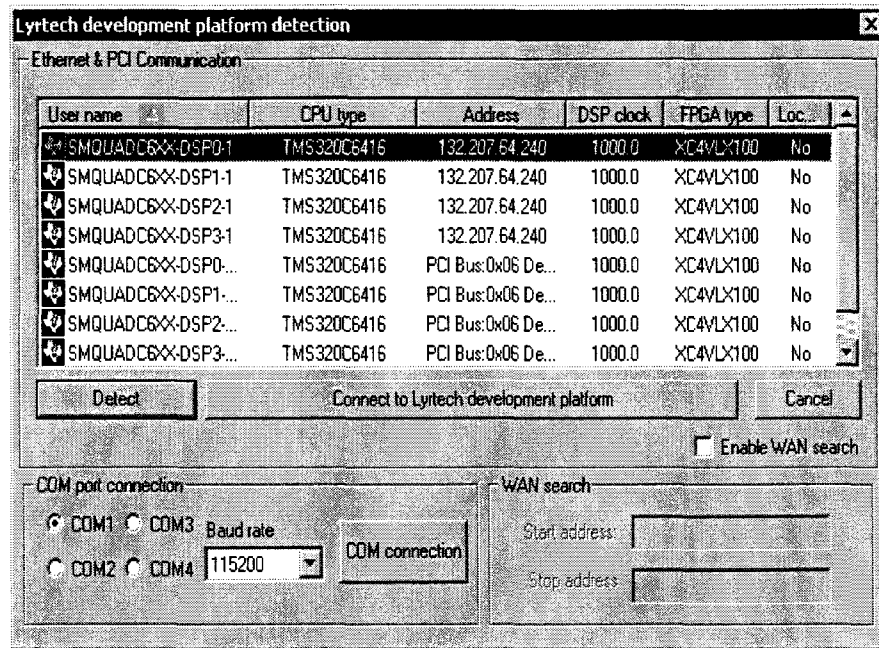


FIGURE 5.4 Interface utilisateur pour la connexion

## 5.2 Le récepteur

La figure 5.5 montre l'architecture du récepteur tel que conçu. Au niveau du VHS-ADC, nous effectuons, tel qu'illustré dans le modèle Simulink présenté à la figure 5.2, le filtrage de Nyquist avant de transmettre les données vers le FPGA du SMQUAD à travers le DRC RapidChannel. Comme nous le constatons à la figure 5.7, le filtre LMS et le recouvrement d'horloge s'effectuent sur ce dernier. Dû à des contraintes de temps, nous n'avons pas effectué la mise en oeuvre d'un PLL, mais ceci devrait se faire sur le FPGA du SMQUAD. Les symboles reçus sont finalement enregistrés dans la mémoire SDRAM et le calcul du taux d'erreurs peut alors s'effectuer hors-ligne.

### 5.2.1 Circuit utilisé au VHS-ADC

Le VHS-ADC au niveau du récepteur contient essentiellement les convertisseurs analogique-numérique et les filtres racine carrée en cosinus surélevé (RCCS). La figure 5.5(a) illustre

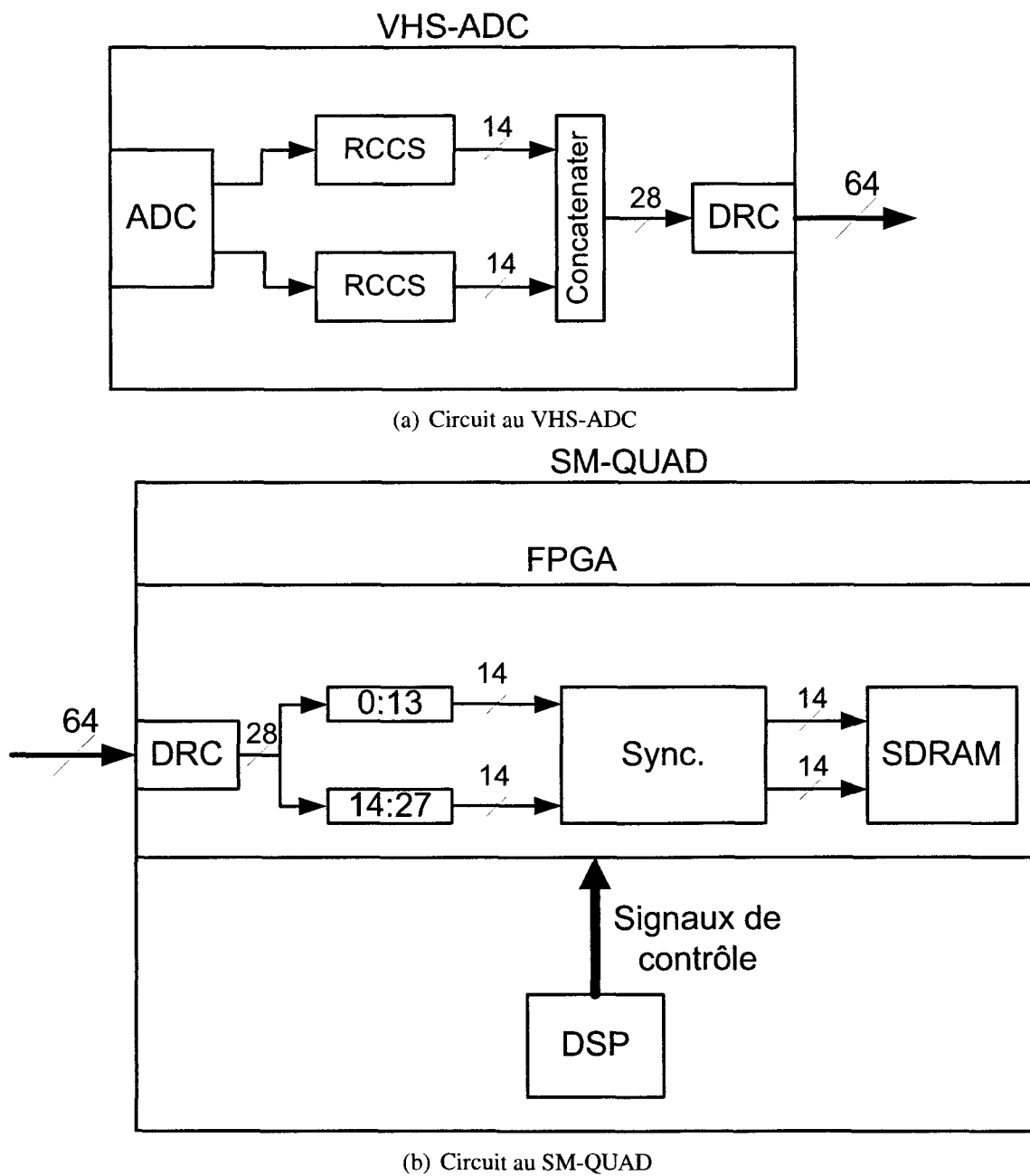


FIGURE 5.5 Architecture du récepteur

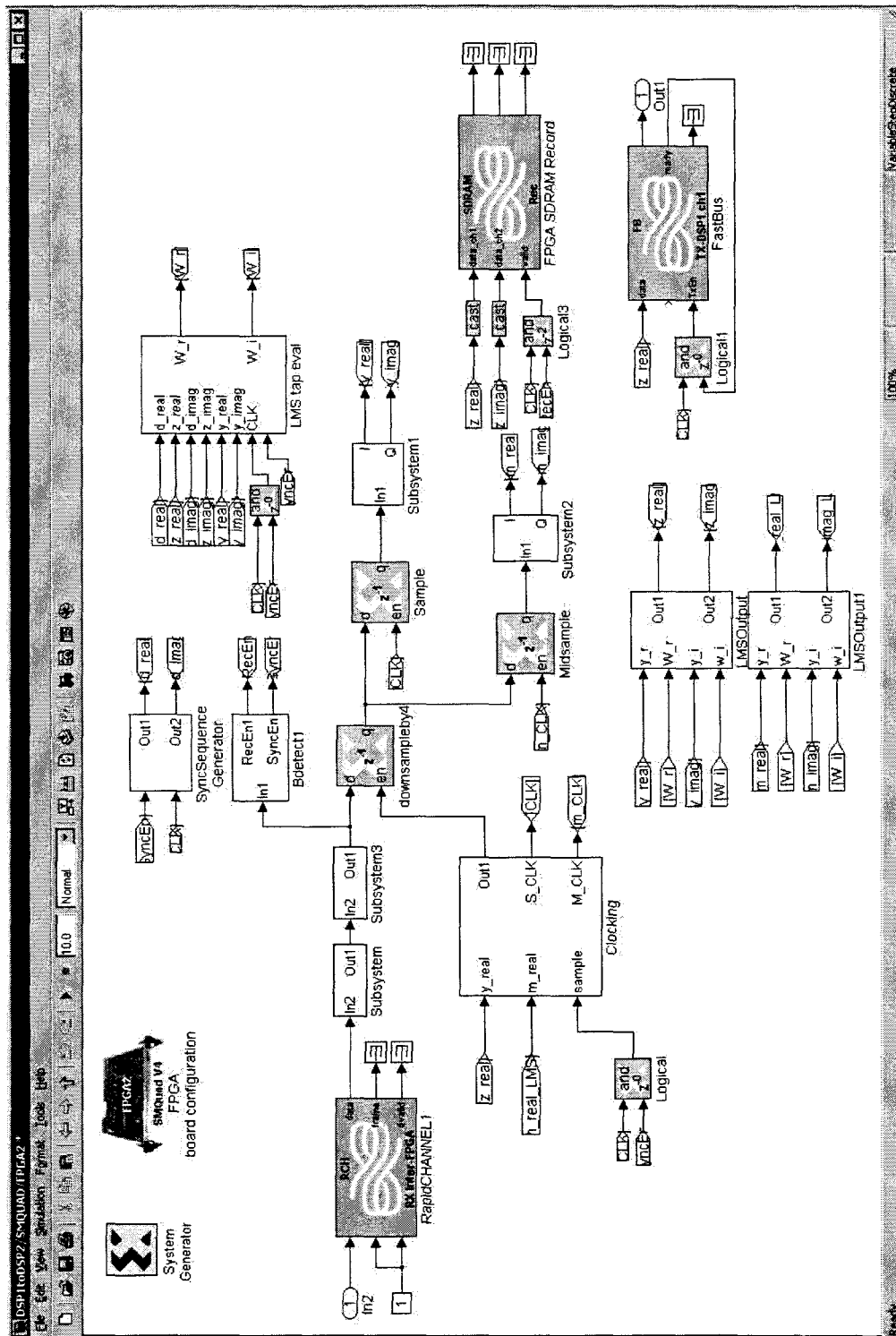


FIGURE 5.6 Modèle Simulink du FPGA du récepteur sur le VHS-ADC

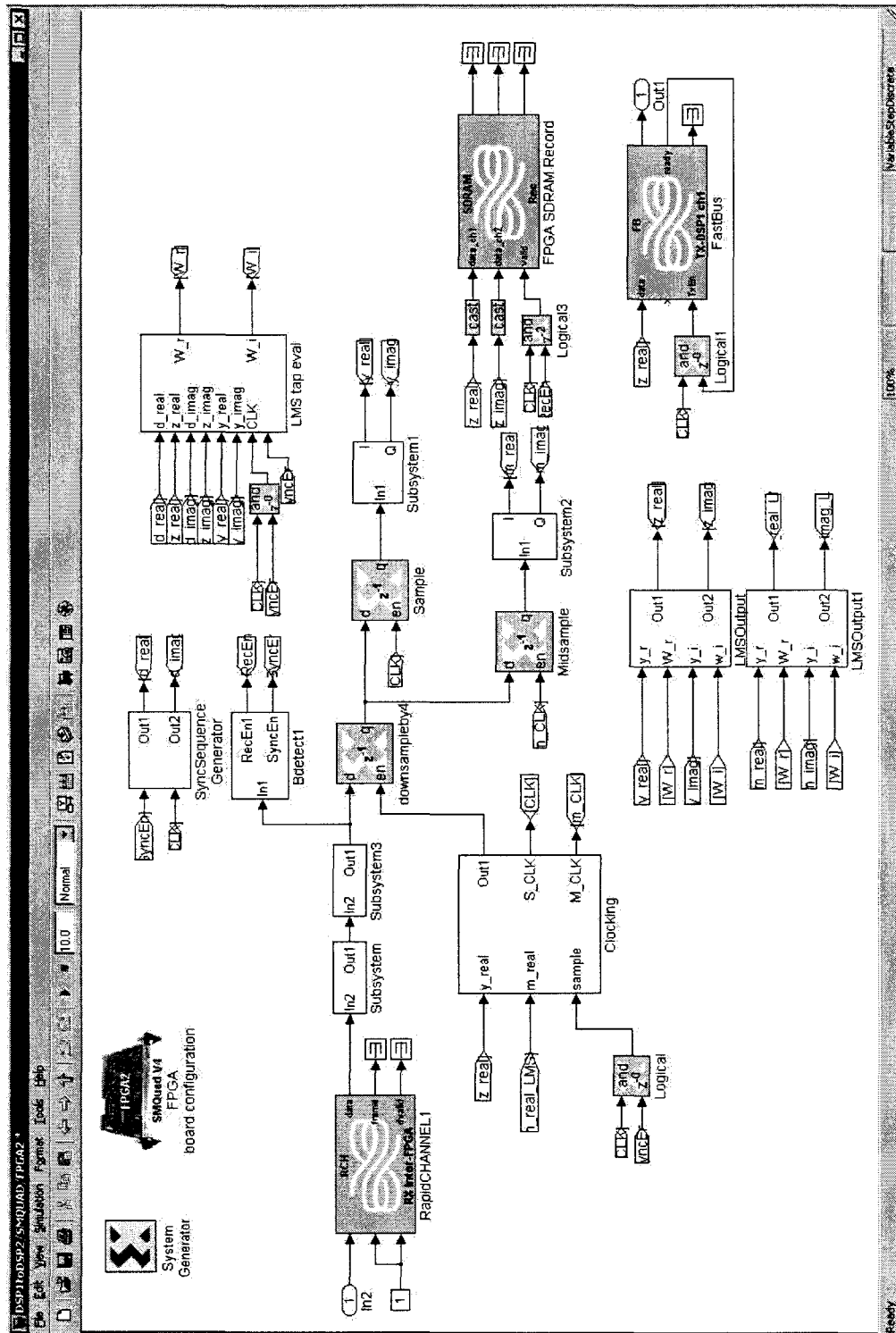


FIGURE 5.7 Modèle Simulink du FPGA du récepteur sur le SMQuad

l'architecture du circuit utilisé.

L'ADC numérise le signal de réception sur 14 bits en point fixe avec une position binaire 0. Les parties réelle et imaginaire du signal sont filtrées séparément à travers les filtres. À noter que pour ces filtres, les paramètres et les coefficients utilisés sont les mêmes que ceux au transmetteur pour en faire des filtres adaptés.

Les sorties de chacun des filtres sont de type point fixe sur 14 bits avec un point décimal à 0 bit. Les sorties sont concaténées en un bus qui sera transmis à travers le RapidChannel du module DRC. Ceci nous permet de transmettre 28 bits de données sur un bus permissible de 64 bits sur le RapidChannel plutôt que de transmettre 14 bits à la fois, ce qui améliore le taux de transmission entre les deux cartes. En d'autres mots, 28 bits de données sont transmis sur un coup d'horloge plutôt que sur deux coups d'horloge.

### **5.2.2 Détection de début de paquet**

Ce circuit a comme entrée le signal filtré. Il permet de détecter le début d'une trame en utilisant des portes logiques et des registres à décalages, et ensuite en comparant la sortie du corrélateur avec des seuils prédéfinis. Ainsi, ce circuit est similaire à celui présenté à la section 3.6. Les sorties sont les signaux RecEn et SyncEn. Ces deux signaux deviennent actifs lorsque le début d'un paquet est détecté. SyncEn reste actif pendant une durée égale à la période de synchronisation. RecEn reste actif pendant l'entière durée d'un paquet. Ceci est illustré à la figure 5.8.

SyncEn est utilisé pour gérer la synchronisation du récepteur. RecEn est utilisé pour fournir une fenêtre temporelle durant laquelle l'enregistrement d'un paquet de données s'effectue.

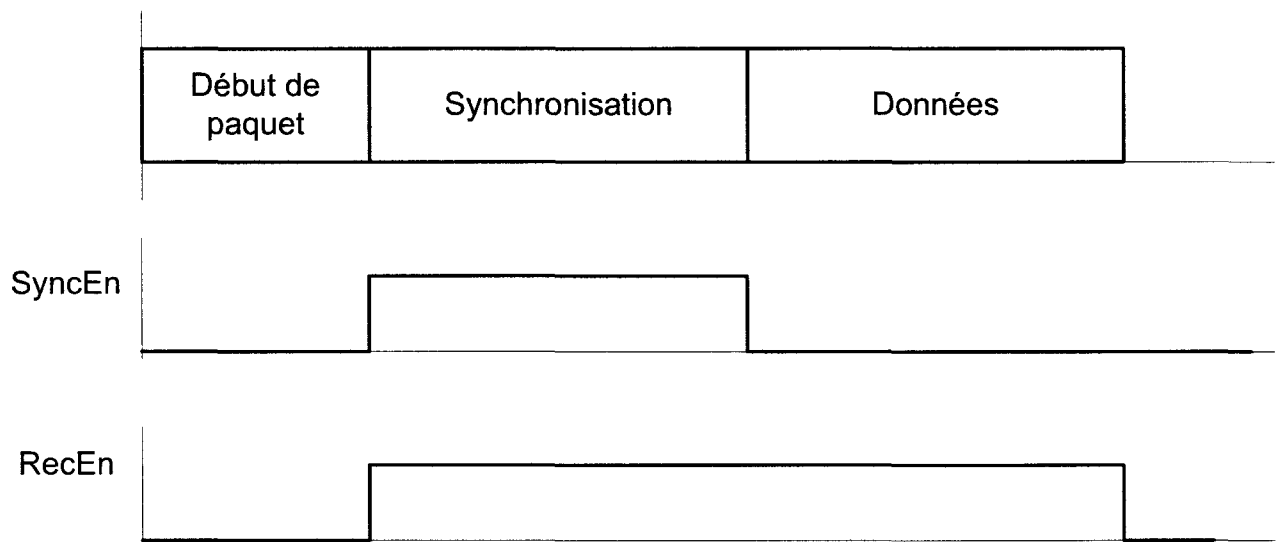


FIGURE 5.8 Diagramme temporel des états de RecEn et de SyncEn

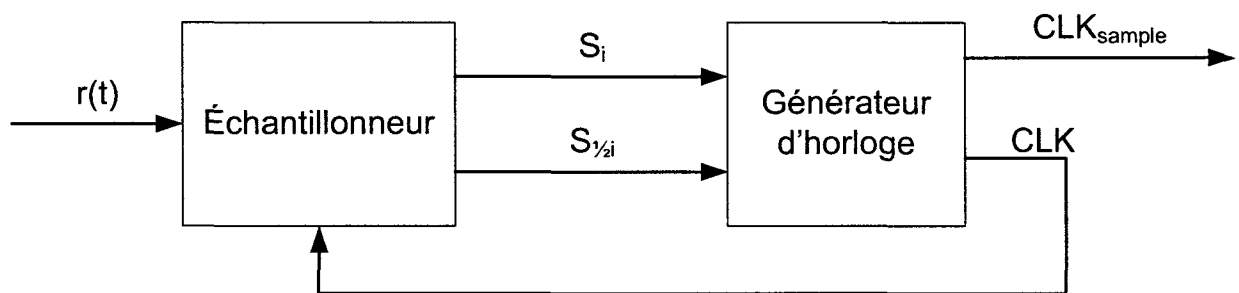
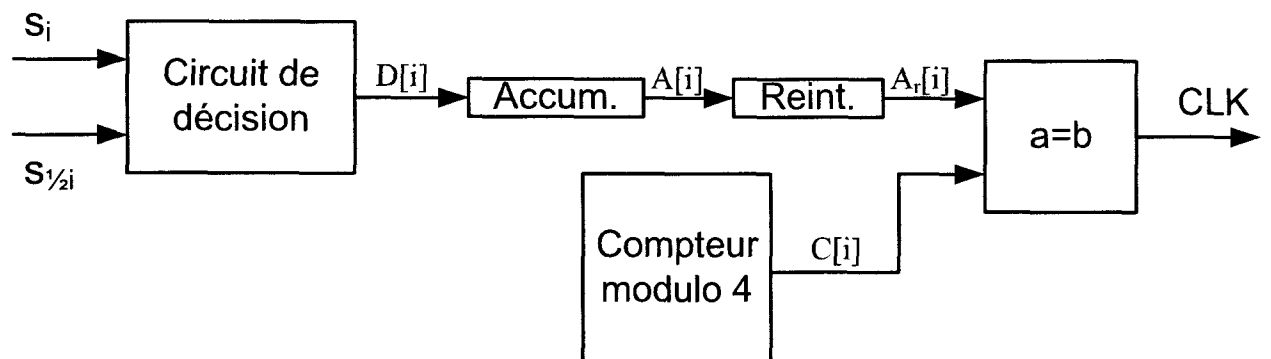


FIGURE 5.9 Schéma bloc du circuit de recouvrement d'horloge

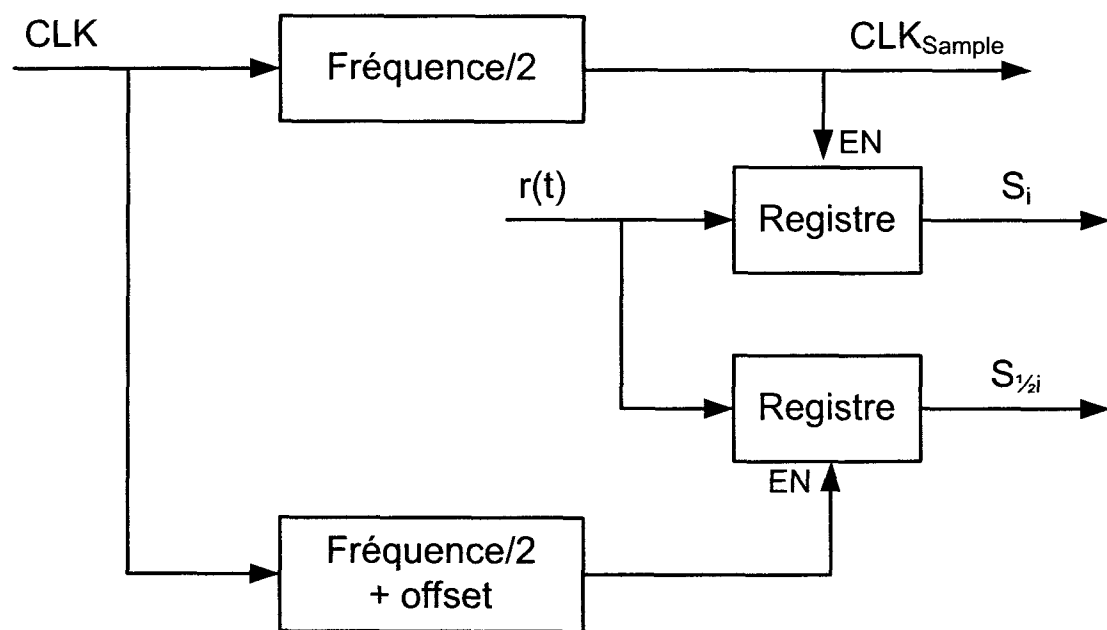
### 5.2.3 Algorithme de recouvrement d'horloge

Le signal reçu est suréchantillonné par un facteur de 8. Grâce à l'algorithme de recouvrement d'horloge, le bon échantillon à la sortie du filtre adapté devra être choisi pour effectuer la démodulation. La figure 5.9 illustre l'architecture que nous avons utilisée pour effectuer ce type de recouvrement.

Le système de sélection d'échantillons est composé de deux éléments. Le premier élément est responsable de la génération d'une horloge à période variable en se basant sur les signes de l'échantillon choisi courant et précédent ainsi que celui de l'échantillon milieu selon l'algorithme décrit à la section 3.7.2. Le bloc constitué du circuit de décision



(a) Circuit bloc du générateur d'horloge



(b) Circuit bloc de l'échantillonneur

FIGURE 5.10 Détail bloc des circuits de recouvrement d'horloge



de la figure 5.10(a) génère un signal  $D[i]$  de valeur  $\pm\delta$  ou nulle, avec  $0 < \delta < 1$ . La sortie de l'accumulateur est, comme son nom l'indique, l'accumulation des décisions. La sortie du bloc  $A[i]$  est arrondie à l'entier près. L'accumulateur est conçu de sorte que la sortie soit représentée par une variable signée de deux bits en complément à 2. Cette dernière est en mode *wrap*. Ceci veut dire que dans le cas où la variable excède la valeur maximale représentable sur 2 bits signés ( $1_{base10} = 01_{base2}$ ), elle devient la valeur minimale ( $-2_{base10} = 10_{base2}$ ). Le bloc de réinterprétation permet de réinterpréter cette valeur en point fixe non-signé. Un comparateur vérifie la valeur du circuit de comparaison avec la valeur d'un compteur binaire modulo 4. Ainsi, un front montant est généré à chaque 4 périodes d'un sous-échantillon (cas 1 de la figure 5.11). Lorsque le circuit de décision émet suffisamment de valeurs négatives, la sortie de l'accumulateur décroît, donc un front montant sera généré après 3 périodes de sous-échantillons (cas 2 de la figure 5.11). Similairement, lorsque le circuit de décision émet suffisamment de valeurs positives, la sortie de l'accumulateur incrémente, donc un front montant sera généré après 5 périodes de sous-échantillon (cas 3 de la figure 5.11).

L'échantillonneur, le second élément du circuit de recouvrement d'horloge, sélectionne les échantillons appropriés du signal de réception. En plus de l'échantillon correspondant au symbole, nous voulons aussi extraire l'échantillon central nécessaire au circuit de décision. La figure 5.10(b) illustre le circuit utilisé pour accomplir cette tâche. L'horloge générée par le circuit de décision étant le double du taux de symboles doit être d'abord ralentie de moitié. Un premier circuit récupère chaque premier front montant pour générer  $CLK_{Sample}$  qui récupère l'échantillon de symbole. Un deuxième circuit récupère le complémentaire, c'est-à-dire chaque deuxième front montant pour sélectionner l'échantillon central. La dynamique de l'échantillonneur est illustrée dans le diagramme temporel de la figure 5.12.

Le reste des éléments se trouvant sur le FPGA fonctionnent à la période d'un sous-échantillon. Cependant, lorsque nous voulons qu'une opération s'effectue à chaque temps

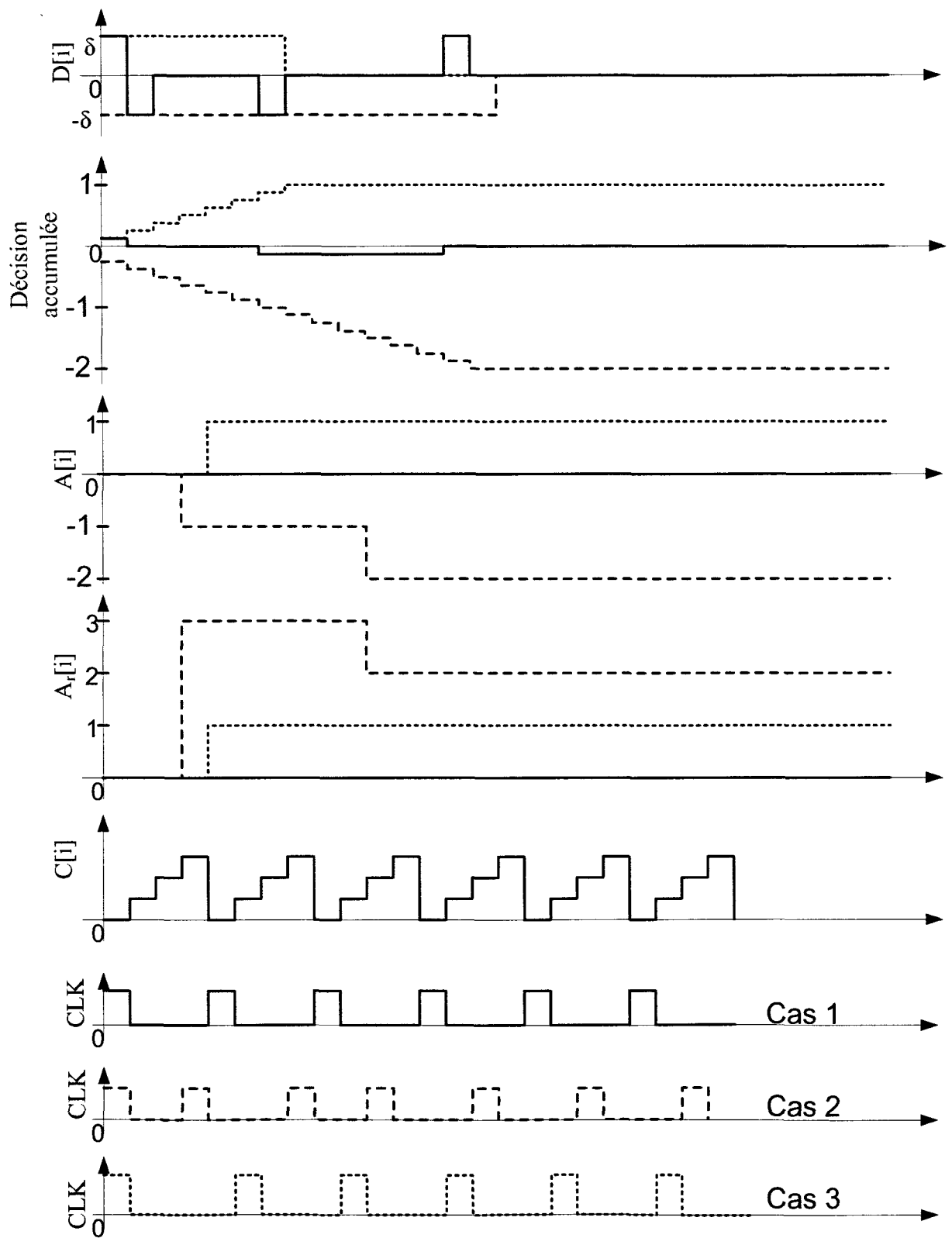


FIGURE 5.11 Diagramme temporel de la génération d'horloge

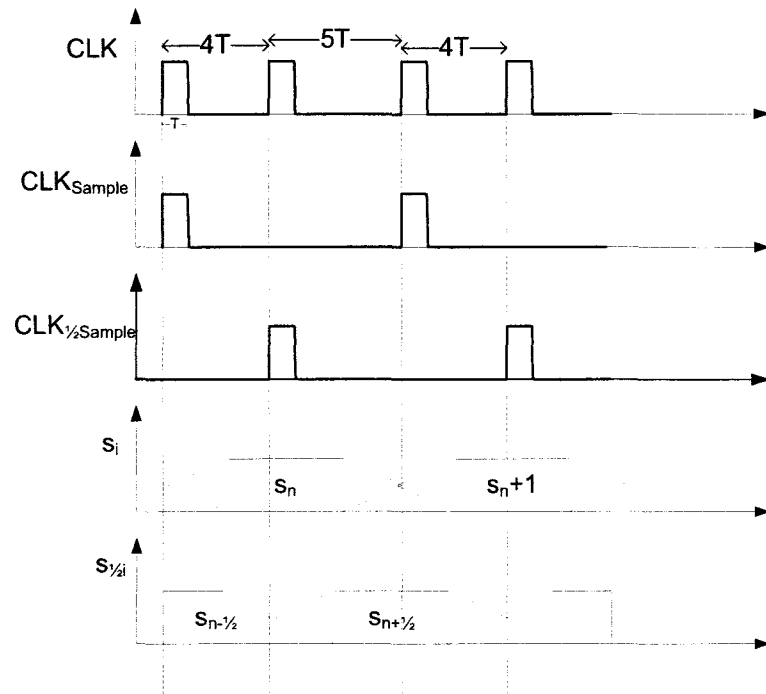


FIGURE 5.12 Dynamique de l'échantillonnage

de symbole, nous utilisons un signal *Enable* qui est activé par le signal  $CLK_{Sample}$ . Par exemple à l'enregistrement des données, nous voulons récupérer un seul échantillon par symbole. Ainsi au port *valid*, le signal  $CLK_{Sample}$  combiné avec d'autres conditions permettra de déterminer si l'échantillon est enregistré.

#### 5.2.4 Filtre d'égalisation

Nous avons séparé le design du filtre d'égalisation en deux éléments : le circuit multiplicateur et le circuit d'estimation de gains. Le circuit multiplicateur effectue l'ajustement de l'amplitude  $A$  et de phase  $\theta$  du signal reçu. Ainsi, le signal reçu est ajusté par un facteur  $Ae^{j\theta} = G_r + jG_i$ , où  $G_r$  et  $G_i$  sont les gains de la partie réelle et imaginaire

respectivement selon l'équation suivante :

$$\begin{aligned}
 s[n] &= (y_{real} + jy_{imaginary}) \cdot Ae^{j\theta} \\
 &= (y_{real} + jy_{imaginary}) \cdot (G_r + jG_i) \\
 &= (y_{real}G_r - y_{imaginary}G_i) + j(y_{real}G_i + y_{imaginary}G_r)
 \end{aligned} \tag{5.1}$$

À partir de l'équation 5.1, nous avons déduit que le circuit multiplicateur complexe serait celui de la figure 5.13. Le gain est de type point fixe représenté sur 10 bits avec le point binaire se trouvant à 5 bits. Par contre, les signaux d'entrée et de sortie sont de type point fixe représentés sur 14 bits avec le point binaire se trouvant à 0 bits. La sortie des multiplicateurs est arrondie en format Fix\_13\_0. Chaque multiplicateur sature sa sortie à sa valeur maximale lorsque celle-ci excède la valeur représentable.

Le circuit d'ajustement de gain est illustré à la figure 5.14. Ce circuit réalise l'équation 3.8 décrite au chapitre 3. Le signal de comparaison  $d$  (la séquence de synchronisation) est préprogrammé dans une mémoire ROM et un compteur permet de définir l'adresse lue pour extraire les données séquentiellement. Le circuit est mis en activation lorsque le signal SyncEn du circuit de détection de paquets est actif. L'accumulateur met à jour sa sortie à chaque coup d'horloge provenant du circuit de synchronisation d'horloge.

### 5.2.5 Enregistrement des données reçues

En référence à la figure 5.5(b), nous avons conçu le récepteur de sorte à enregistrer des blocs de données reçues. Puisque chacun des canaux de la mémoire vive disponible sur le FPGA peut contenir une donnée de 16 bits au plus, nous avons utilisé deux canaux pour enregistrer les 14 bits de la partie réelle et imaginaire du signal de réception séparément. La donnée à enregistrer est valide lorsque les signaux  $CLK_{Sample}$  et  $RecEn$  sont à 1. Ainsi nous nous assurons d'enregistrer seulement un échantillon pour chaque sym-

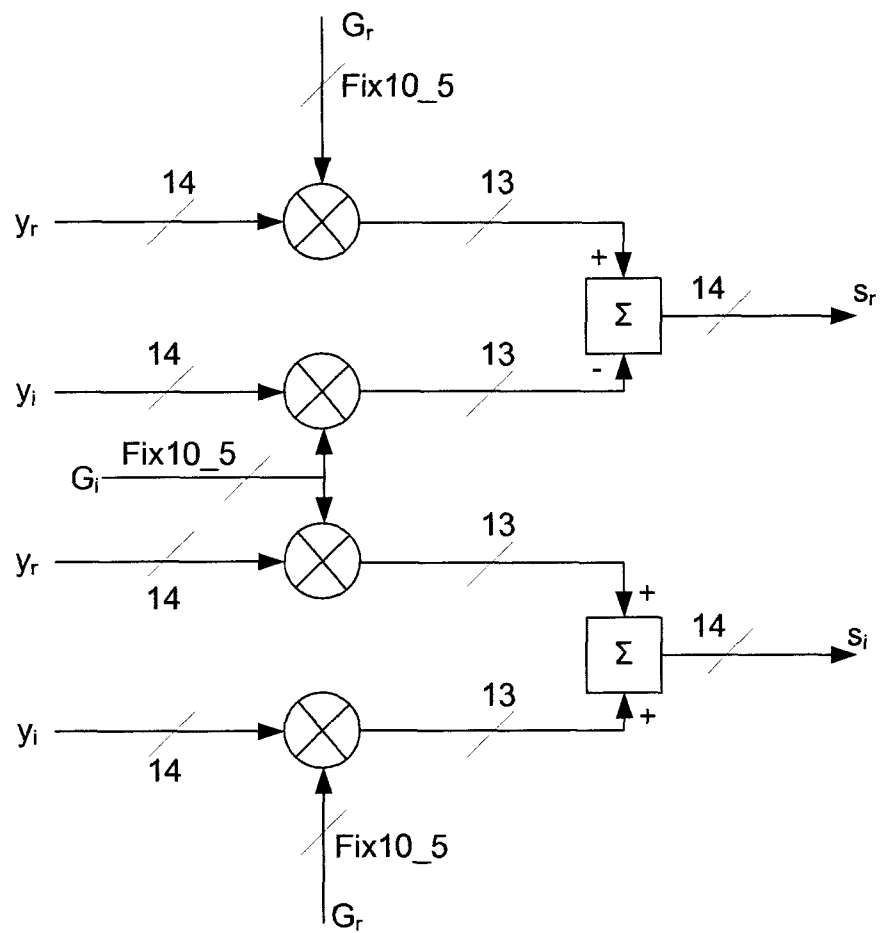


FIGURE 5.13 Circuit de multiplication complexe

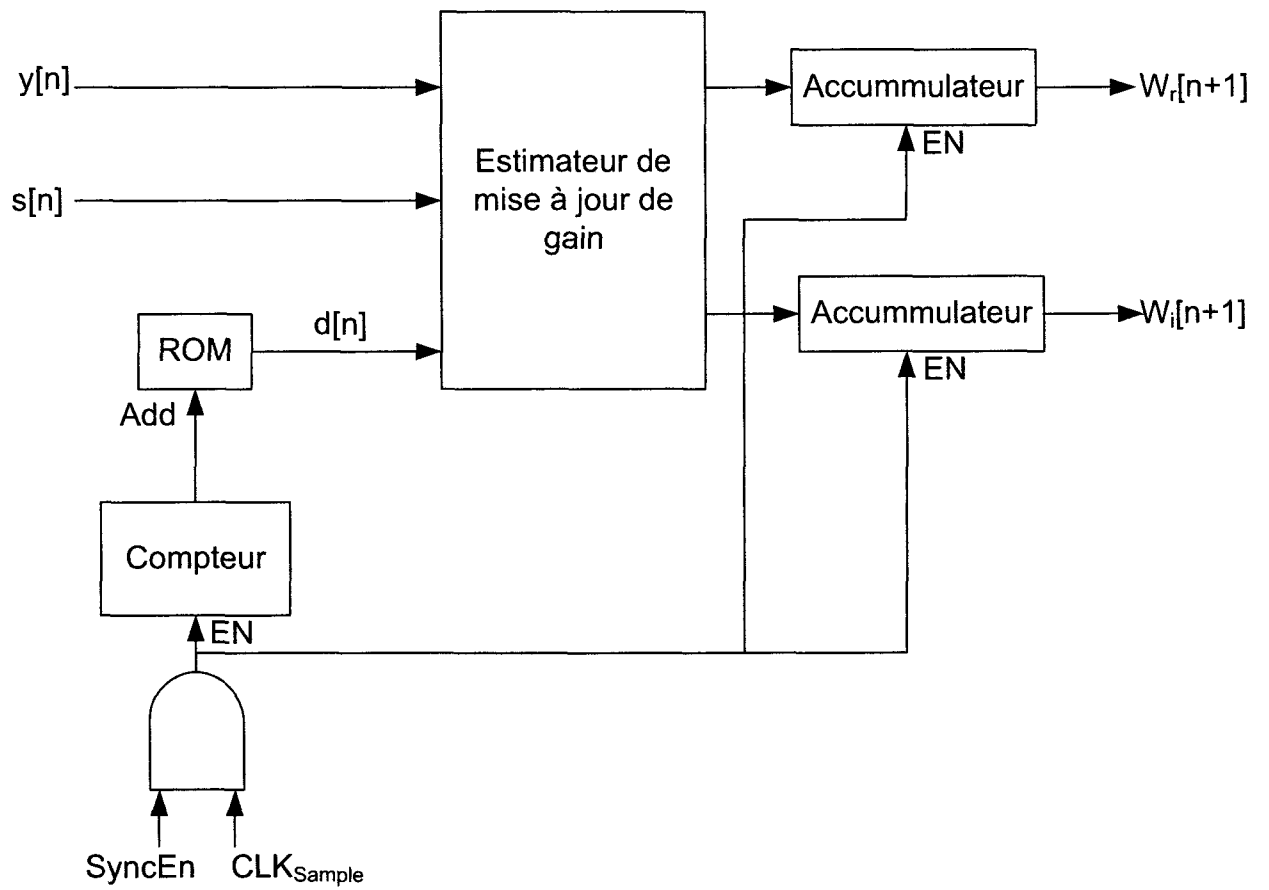


FIGURE 5.14 Circuit d'ajustement du gain du filtre LMS

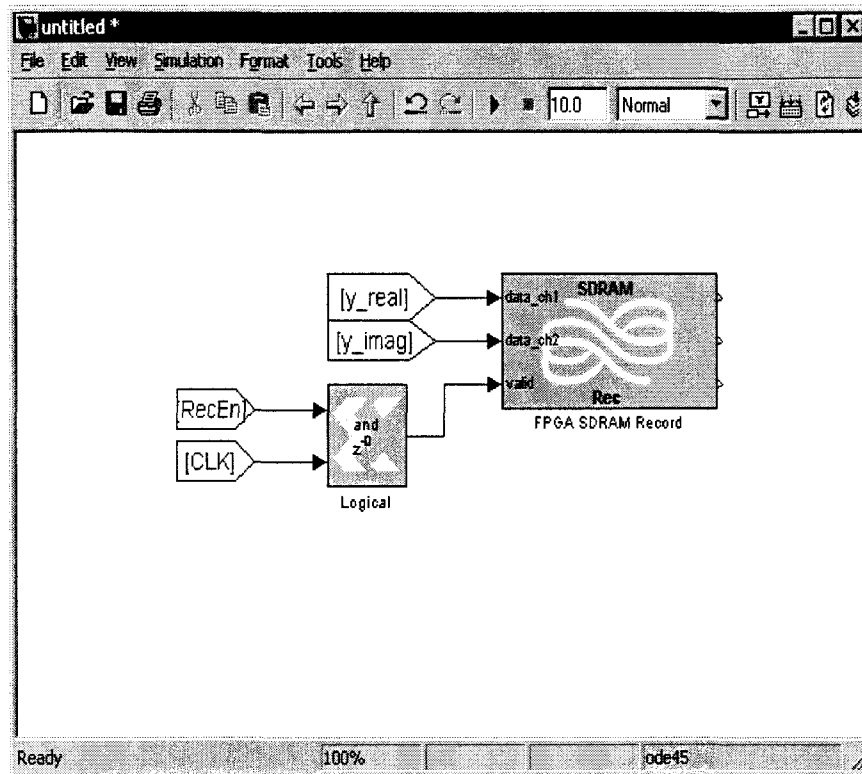


FIGURE 5.15 Logique de sélection des données enregistrées

bole reçu. Lors de chaque enregistrement, nous avons récupéré 8192 échantillons. Nous nous sommes aussi limités à ce nombre d'échantillons puisque nous avons constaté une instabilité lorsqu'il est trop élevé.

L'enregistrement est initié par l'utilisateur à partir du DSP, tel que décrit à la section 4.4.

### 5.2.6 Blocs DSP

Les blocs utilisés pour le récepteur au niveau du DSP sont montrés à la figure 5.16. Dans le design actuel, le DSP n'effectue aucune opération de traitement de signal. Comme nous l'avons indiqué dans les sections précédentes, le traitement de signal est effectué au niveau des FPGA. Le reste des opérations, tel que le calcul de taux d'erreurs, est effectué hors-ligne. Le DSP est donc utilisé seulement à des fins de contrôle. Ainsi, nous y spécifions

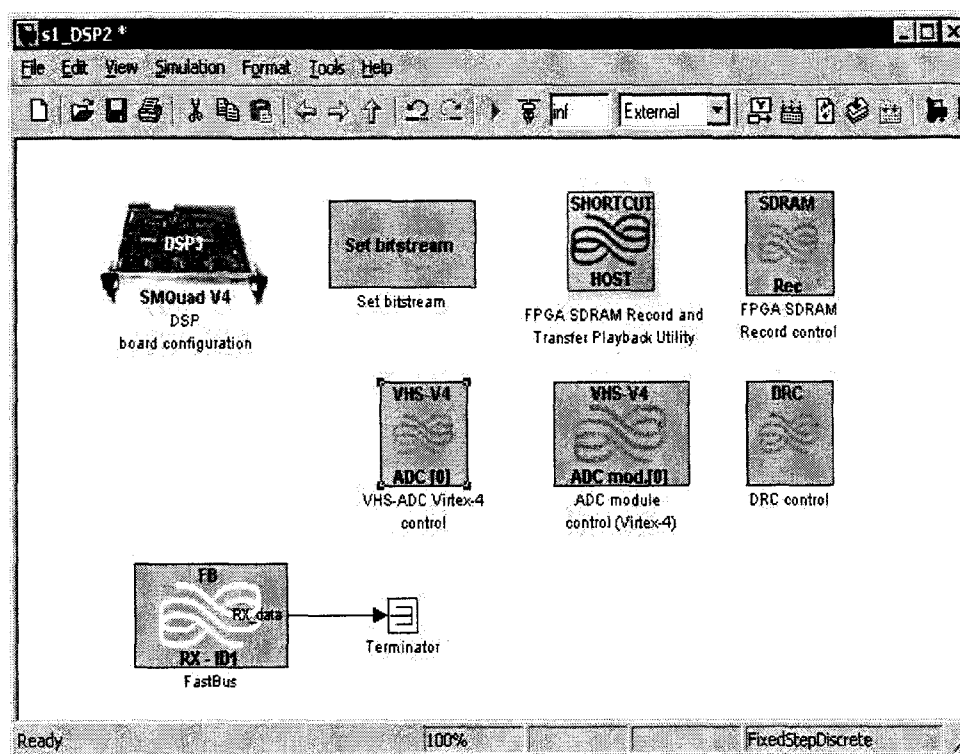


FIGURE 5.16 Blocs utilisés au DSP du récepteur

les paramètres à être utilisés par l'ADC et le module DRC. Nous y effectuons aussi le contrôle de l'enregistrement de données.

Un point particulier à noter est que nous avons inséré un canal fast-bus en réception. En réalité, ce canal ne transfère aucune donnée utile. Il a été ajouté car nous avons constaté qu'en son absence, le DSP bloquait. Nous n'avons pas enquêté davantage sur ce problème.

### 5.3 Performances

Tout au cours de la conception de ce système de communications, nous avons dû procéder par étapes et effectuer une série de tests afin de valider la fonctionnalité de chacun des blocs avant d'en arriver à l'ensemble. Dans cette section, nous décrivons trois des étapes clefs qui nous ont permis de caractériser le système.



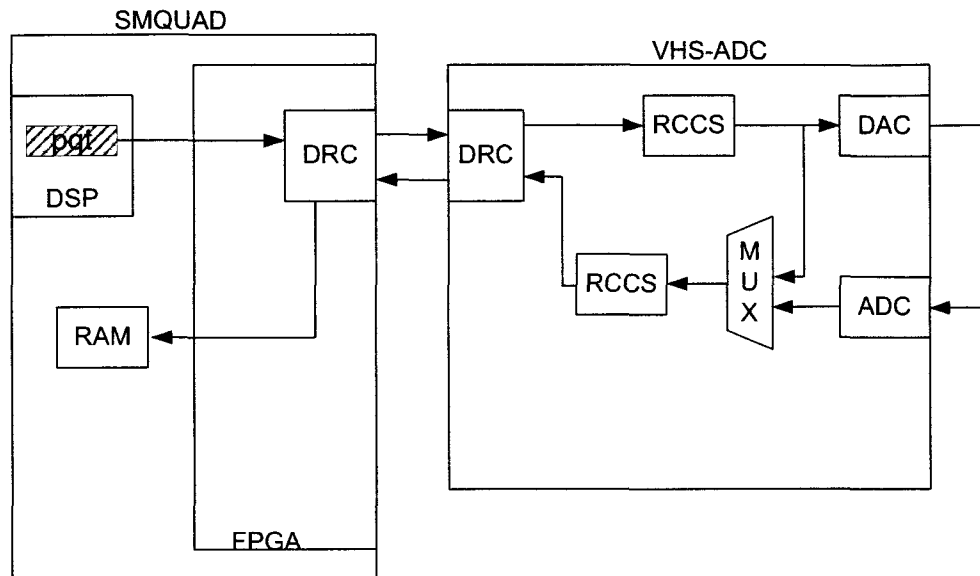


FIGURE 5.17 Circuit pour la caractérisation du transmetteur

### 5.3.1 Caractérisation du transmetteur

Afin de tester le transmetteur, nous nous sommes basés sur le circuit de la figure 5.17 permettant d'effectuer une boucle interne ou externe. Nous avons utilisé le diagramme d'oeil et la constellation des symboles reçus pour observer et mesurer certaines caractéristiques du transmetteur.

#### Diagramme d'oeil pour la boucle interne

Par référence à la figure 5.17, c'est le circuit qui passe à travers les deux filtres de Nyquist sans sortir de la plateforme. Nous avons surtout utilisé cette boucle pour vérifier que les filtres de Nyquist sont effectivement complémentaires de sorte à avoir un niveau d'ISI faible. La figure 5.18 montre les résultats obtenus pour la partie réelle et imaginaire du signal. Nous en faisons les observations et déductions suivantes :

- Au point ayant le plus grand écart, il n'y a pas beaucoup de variations du signal. Nous en concluons que le niveau de bruit rémanent est faible.

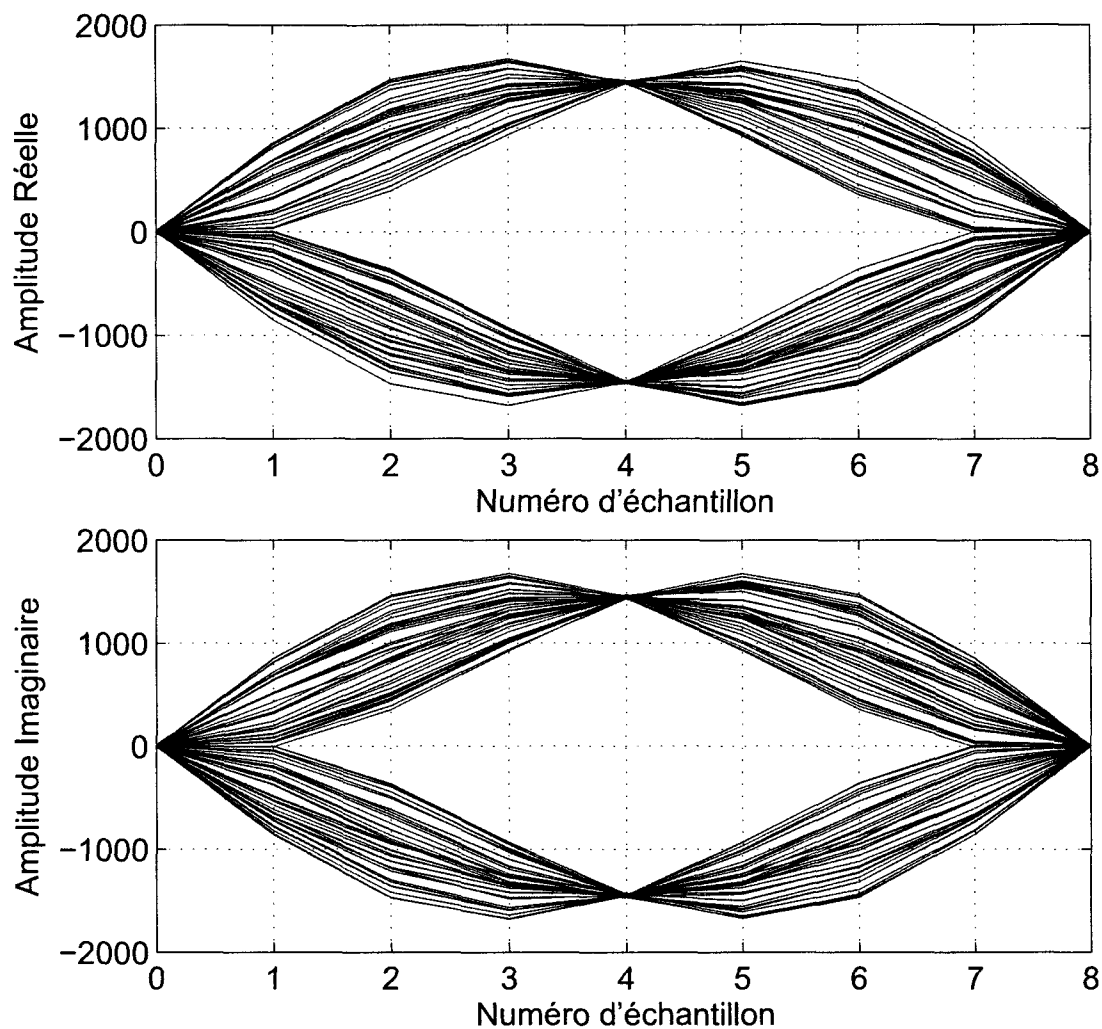


FIGURE 5.18 Diagrammes d'oeil pour une boucle interne

- Nous considérons que l'ouverture de l'oeil est suffisamment grande. Donc, la sensibilité de ce système aux erreurs de synchronisation est acceptable.

### Diagramme d'oeil pour la boucle externe

Par référence à la figure 5.17, c'est le circuit qui passe à travers les deux filtres de Nyquist avec un chemin de sortie de la plateforme passant par les deux DAC et un chemin de retour passant par les ADC. La figure 5.19 montre les résultats obtenus pour la partie réelle et imaginaire du signal. Nous pouvons faire les observations suivantes :

- Le bruit au point de plus grand écart de l'oeil a augmenté. Cependant, ces oscillations sont faibles par rapport à l'ouverture de l'oeil. En effet, les oscillations sont à peine de 10% de l'ouverture, indiquant qu'à pleine puissance de transmission, ce bruit n'influencera pas beaucoup le BER.
- Le point ayant la plus grande ouverture est légèrement en retard par rapport à l'échantillon dont nous avons pris les mesures (quatrième échantillon). Donc, le système ne sera pas en mesure d'échantillonner au meilleur moment. Ceci est dû au fait que le délai introduit par le système n'est pas un multiple du temps d'échantillonnage.
- L'ouverture de l'oeil n'a pas beaucoup augmenté et reste à une valeur jugée acceptable. Donc, nous pensons que la sensibilité restera acceptable.

### **Constellations**

Nous avons aussi tracé la constellation du signal reçu pour chacun des deux cas. En se basant sur la figure 5.20, nous pouvons affirmer que les symboles reçus n'ont aucun déphasage. Nous pouvons aussi faire les observations suivantes :

- Le signal ayant traversé la boucle externe est plus bruité que celui ayant passé par la boucle interne. Le SNR en boucle interne est de 24 dB alors que le SNR en boucle externe est de 18 dB. Ceci est un résultat attendu. Nous pensons que le bruit de quantification aux niveaux des DAC ainsi que l'interférence entre symboles en sont les principaux coupables.
- Le signal ayant traversé la boucle externe a été atténué à travers les câbles reliant les DACs aux ADCs. Cette atténuation ne semble pas être majeure à priori.
- Il existe un DC au niveau du signal reçu par la boucle externe. Les résultats vus ici montre que ce DC est de l'ordre de 100 à 200 unités. Nous avons malheureusement observé au courant des mesures que la valeur de ce DC n'est pas constante. Afin de pouvoir effectuer les mesures au niveau du récepteur en boucle externe, un algorithme pour éliminer ce DC sera requis.

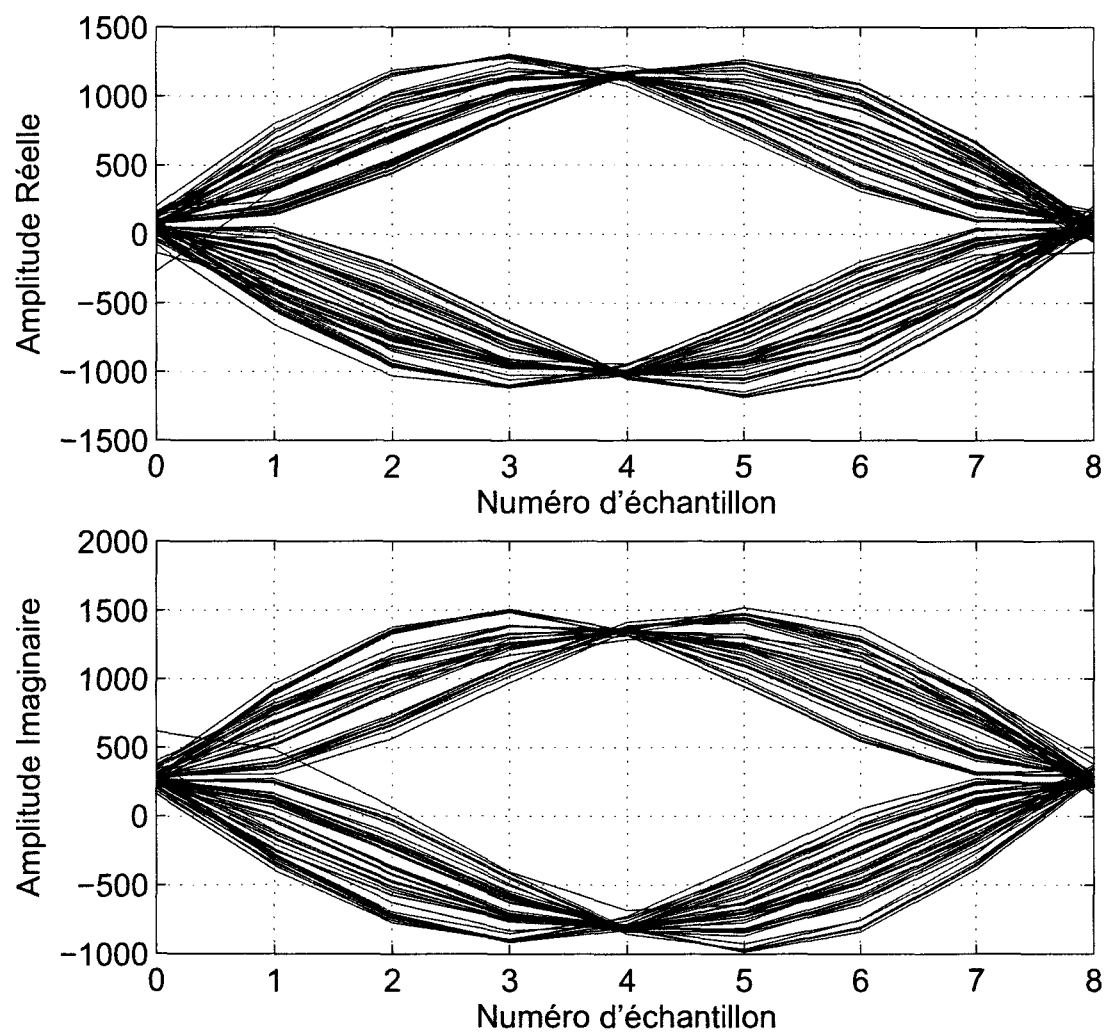
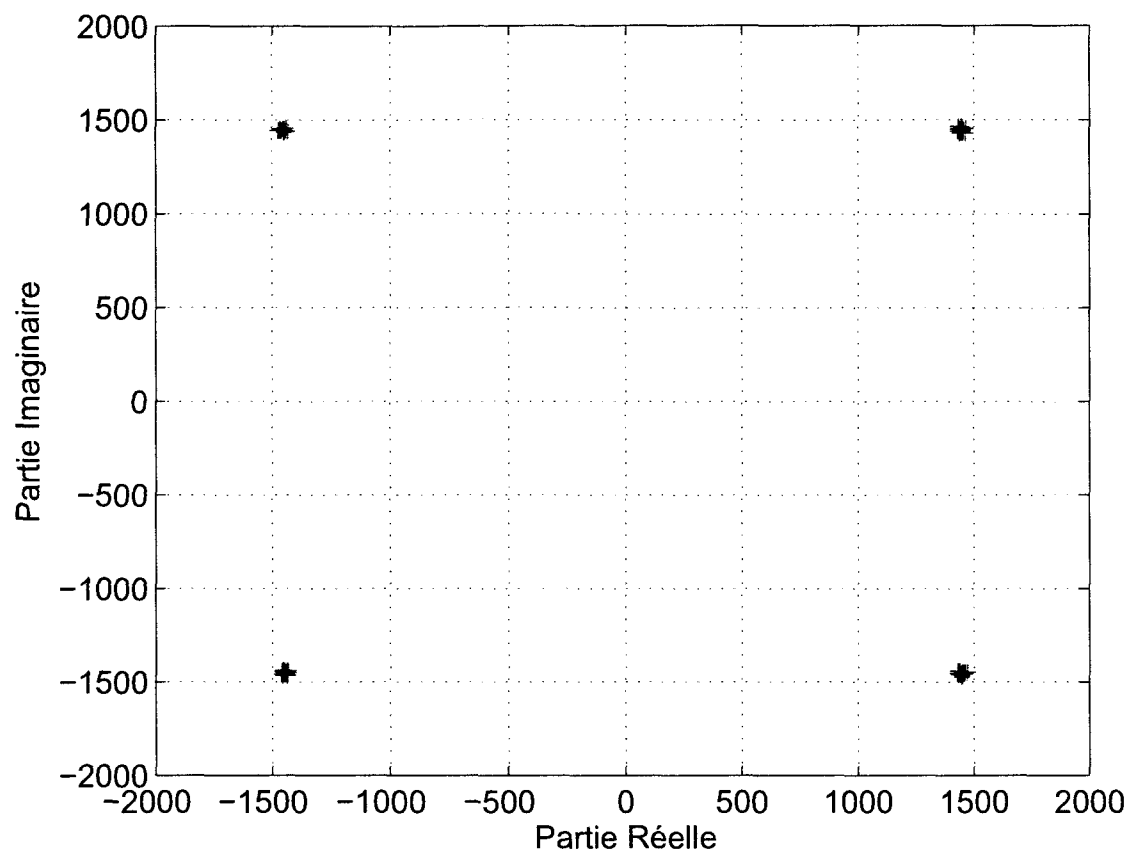
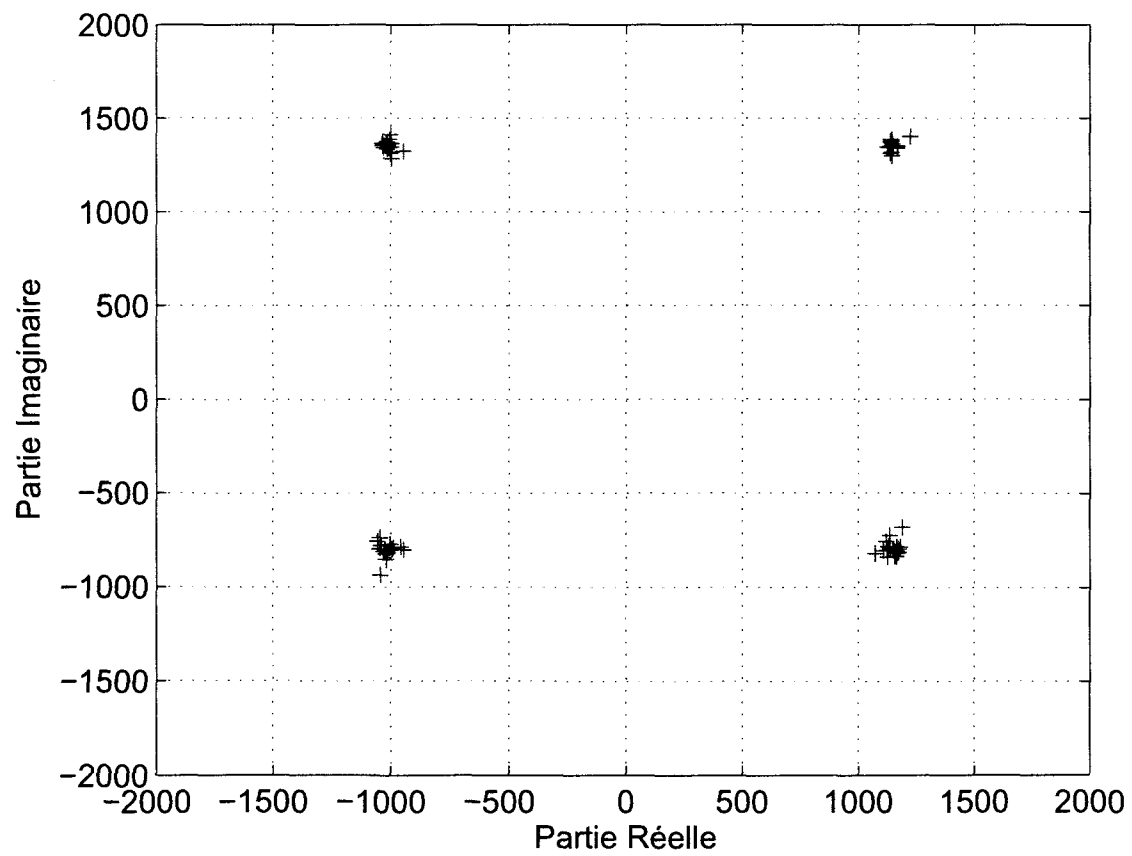


FIGURE 5.19 Diagrammes d'oeil pour une boucle externe



(a) Boucle interne



(b) Boucle externe

### 5.3.2 Vérification de la fonctionnalité du récepteur

Il est évident qu'avant de procéder aux mesures de performance, nous devons confirmer que le récepteur est fonctionnel. Nous avons donc procédé à la transmission de données selon les deux chemins illustrés à la figure 5.21.

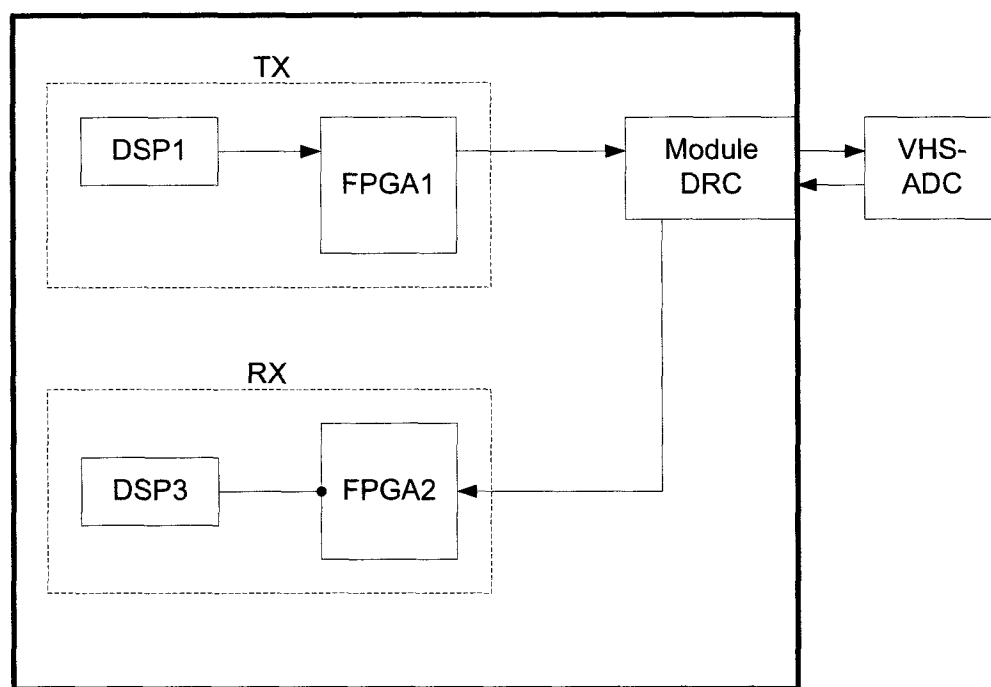
Nous avons constaté que dans chacun des deux cas, les données transmises sont effectivement reçues. Cela signifie que le bloc de détection de paquets et les boucles de synchronisation sont fonctionnels. À noter que nous avons effectué la vérification de fonctionnalité à haut SNR. À bas SNR, nous avons constaté que le DC présent dans la boucle externe provoque des détériorations importantes aux algorithmes de synchronisation.

### 5.3.3 Performance du récepteur

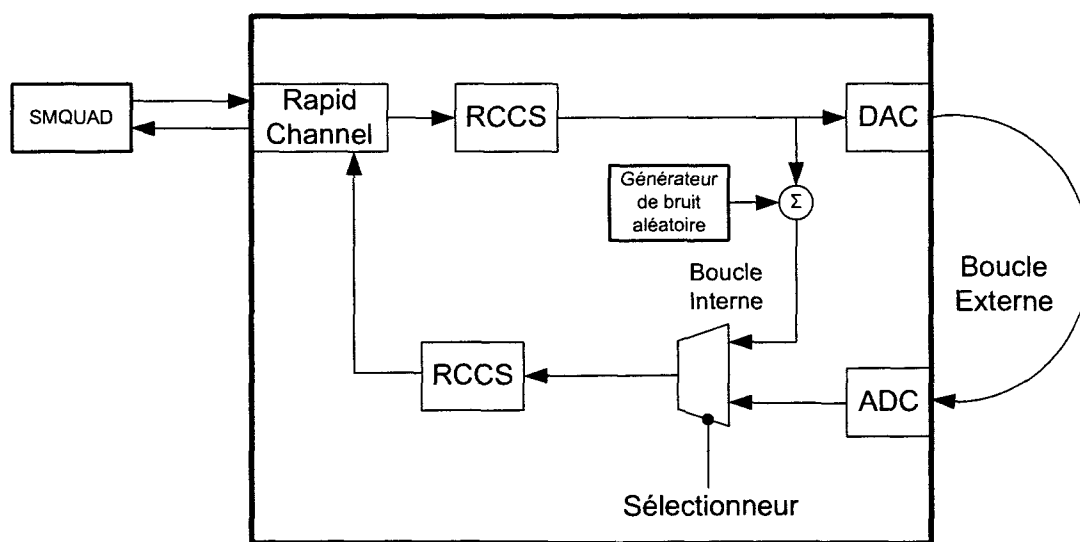
Afin de quantifier la performance en boucle interne du récepteur, nous avons tracé la courbe du taux d'erreurs binaires en fonction du SNR du signal au récepteur. De toute évidence, nous devons pouvoir faire varier et de mesurer le SNR du signal de réception pour effectuer les mesures de nombre d'erreurs. De plus, afin d'avoir des résultats de bonne précision, il est nécessaire de mesurer suffisamment d'échantillons.

#### Variation du SNR

La variation du SNR peut s'effectuer en faisant varier la puissance du signal émis. Pour ce faire, nous avons fait varier l'amplitude de chaque symbole du paquet transmis à sa génération. Par ailleurs, il est utile d'être en mesure de définir les conséquences de différents niveaux d'atténuation sur le SNR. En considérant une puissance de bruit  $N_0$  qui ne varie pas, nous pouvons affirmer, en consultant l'équation 5.2, que pour diminuer le SNR de



(a) SMQUAD



(b) VHS-ADC

FIGURE 5.21 Architecture utilisée pour la vérification du récepteur

$xdB$ , le signal doit être atténué de  $10^{0.1x}$ .

$$\begin{aligned} SNR_{dB} &= 10 * \log \left( \frac{KP}{N_0} \right) \\ &= 10 \log(K) + Constante \end{aligned} \quad (5.2)$$

Ainsi, dans un premier temps, nous avons calibré notre signal pour avoir un point d'opération jugé acceptable d'environ  $13dB$  de SNR. À un tel point d'opération, le taux d'erreurs binaires est de l'ordre de  $10^{-5}$  à  $10^{-6}$ . Par la suite, le signal est davantage atténué pour pouvoir faire les différentes mesures désirées.

### Estimation du SNR

Il est nécessaire de déterminer le SNR du signal de réception. Nous sommes donc intéressés à déterminer la puissance moyenne du signal de réception, ainsi que le bruit. Le signal reçu comporte une composante DC. Donc, dans un premier temps, nous avons éliminé cette composante et estimé la puissance du signal.

$$\begin{aligned} P &= E[\Re(r(t) - E[r(t)])^2 + \Im(r(t) - E[r(t)])^2] \\ &= E[\Re(r(t))^2] + E[\Im(r(t))^2] - E[\Re(r(t))]^2 - E[\Im(r(t))]^2 \end{aligned} \quad (5.3)$$

La puissance du bruit peut être estimée par l'équation 5.4.

$$\begin{aligned} \sigma_n^2 &= E \left[ \Re\{r(t)\} - \left( E[\Re(r(t))] \cdot \text{sign}(r(t)) \right) \right]^2 \\ &\quad + E \left[ \Im\{r(t)\} - \left( E[\Im(r(t))] \cdot \text{sign}(r(t)) \right) \right]^2 \end{aligned} \quad (5.4)$$

Nous constatons ici l'hypothèse que dans le calcul du SNR, nous faisons abstraction du DC présent dans le signal. En effet, nous avons assumé que lorsque sa valeur est faible,



il ne devrait pas influencer de façon aléatoire le BER. Par contre, si elle est trop élevée par rapport au niveau de puissance du signal, le DC devra être pris en considération ou le design devra être corrigé pour l'éliminer.

## Résultats

Nous avons répétitivement transmis plusieurs paquets. Pour chaque groupe de paquets transmis, nous avons évalué le BER en comparant les données reçues avec celles transmises. Nous avons aussi évalué le SNR des différents paquets reçus. À noter que l'équipement disponible permettait de récolter un maximum d'environ 15,000 échantillons dans chaque enregistrement. Il nous a alors fallu estimer le taux d'erreurs à haut SNR en combinant les résultats de plusieurs paquets récoltés. À la figure 5.22, nous observons les différentes mesures obtenues. Comme le bruit est un processus aléatoire, nous observons que ces points sont distribués selon une certaine tendance.

Sachant que le taux d'erreurs binaires varie de façon générale en fonction de  $Q(\sqrt{SNR})$ , nous pouvons, à haut SNR affirmer que l'approximation  $BER = ae^{b \cdot SNR}$  est raisonnable. Ceci nous permet de tracer la courbe de régression à travers les divers points collectés tel qu'observé à la figure 5.22.

Nous pouvons conclure que le système conçu suit la performance attendue avec une dégradation inférieure à  $0.5dB$  par rapport à la courbe théorique. Cette dégradation est particulièrement significative à haut SNR mais demeure acceptable.

Il existe un écart entre les points mesurés et la courbe de régression. Cette écart provient du fait que le BER était mesuré sur une quantité limitée de paquets. Pour être en mesure de rapprocher les mesure de la courbe, davantage de nombre de symboles sont nécessaires pour chaque valeur de SNR.

De plus, nous constatons qu'à haut SNR, le BER est comparativement grand par rapport

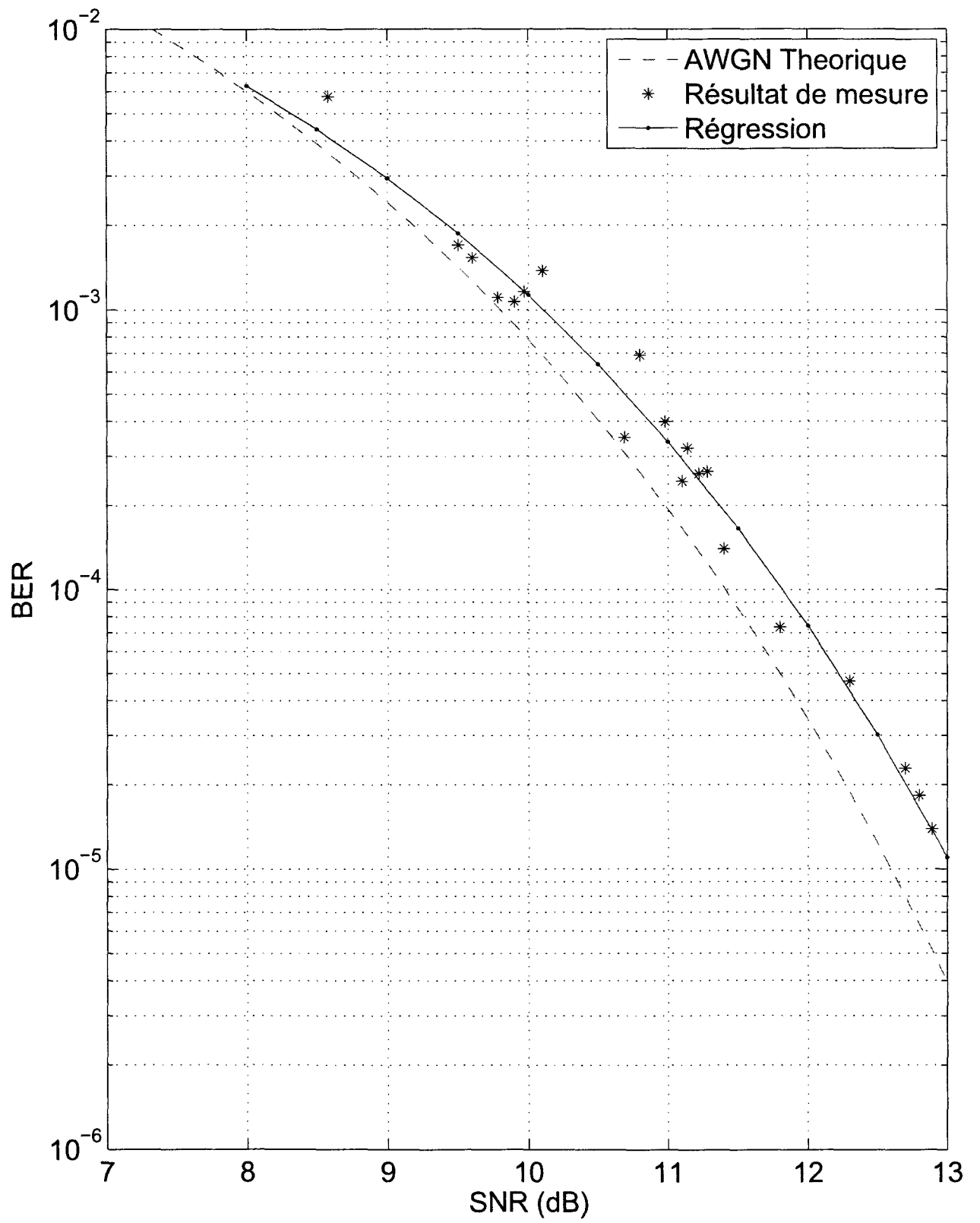


FIGURE 5.22 Courbe de BER vs SNR

au BER à bas SNR. Ceci s'explique par le fait qu'à haut SNR, un plus grand nombre de symboles doivent être mesurés pour avoir des résultats précis puisque les erreurs binaires se font rares (de l'ordre de 1 erreur pour chaque 10,000 symboles transmis).

## CHAPITRE 6

### DISCUSSIONS ET CONCLUSION

#### 6.1 Conclusion

Le développement de ce système de communications sans-fil point à point s'est montré relativement complexe et rempli de défis. Comme nous l'avons mentionné au chapitre 1, il est nécessaire avant tout de définir le type de communications à implémenter afin de comprendre les enjeux. Dans le cadre de notre projet, nous avons décidé d'utiliser le système le plus simple, notamment un canal à évanouissement lent et non-sélectif en fréquences.

La modélisation du système à concevoir nous a permis de valider les différents algorithmes que nous avons projeté d'utiliser par la suite. Elle nous a aussi permis d'avoir une idée quantitative du type de performance à laquelle nous pouvons nous attendre avec un tel système.

La première phase d'implémentation du système indique que nous sommes sur le bon chemin. Les performances obtenues sont acceptables. Avec l'ajout d'une boucle PLL, nous nous attendons à une légère dégradation de la performance. Il y a cependant plusieurs améliorations qui peuvent être faites, mais les perspectives de développement sont bonnes.

#### 6.2 Améliorations

Le but de ce projet était de faire le premier pas pour avoir un système fonctionnel et pour comprendre les outils proposés par Lyrtech Inc. Cependant, nous avons noté plusieurs

améliorations qui peuvent être apportées au système. Certaines d'entre elles se situent au niveau du design effectué, d'autre au niveau de l'équipement disponible.

### **Améliorations du design**

Nous avons noté les points suivants :

- L'élément PLL manquant dans l'implémentation hardware mérite d'y être ajouté. Afin de tester cela, il serait judicieux d'utiliser une communication entre deux plateformes plutôt qu'une boucle.
- Il serait convenable d'effectuer une estimation de l'intervalle de confiance selon un test d'hypothèse pour valider la courbe BER vs SNR tel que mesuré.
- Nous avons eu quelques difficultés à effectuer une estimation en temps réel du taux d'erreur binaires au niveau du DSP. Un tel module était initialement désirable car ceci permettrait de faire l'estimation du taux d'erreurs binaires sur un plus grand nombre d'échantillons.
- Le système fourni ici n'est pas optimisé en aucun sens. Une des possibilités serait de chercher les valeurs des paramètres du récepteur tel que le gain de la boucle de rétroaction du filtre LMS de sorte à ce que la longueur d'entête requise soit minimale.
- Nous avons constaté la présence d'une composante DC dans le signal de réception. Sa source devrait être davantage investiguée pour l'éliminer car nous pensons que cela risque d'influencer les résultats de calcul du taux d'erreurs.

### **Améliorations du fournisseur**

Nous avons noté les points suivants qui devraient être corrigés ou améliorés par le fournisseur :

- Lorsque nous tentons de récolter des données au niveau de la RAM du FPGA, nous avons constaté une instabilité du système. Le système devenait instable et "crashait".

- Il serait intéressant d'avoir une horloge variable qui serait un non-multiple du 104 MHz de l'ADC. Présentement ce dernier est l'horloge que nous utilisons et qui est propagé à travers le système.
- Lorsque nous tentons d'enregistrer les données se trouvant dans la RAM du FPGA, nous avons constaté qu'il fallait démarrer le "SDRAM Record and Playback Transfer Utility" avant de connecter les DSP. Sinon, le système produira des comportements inattendus.
- Nous aimerions pouvoir transférer les données vers le DSP et effectuer l'enregistrement temps-réel vers un fichier sur le disque dur. Ceci a été proposé à Lyrtech Inc., mais actuellement il n'est pas encore disponible.
- Nous avons constaté que le bloc "Playback from RAM" fourni dans les librairies de Lyrtech Inc. n'est pas fonctionnel. En effet, le modèle ne compile pas lorsque ce bloc est présent. Ceci est présentement en investigation auprès du fournisseur.

### 6.3 Perspectives

Ce projet marque un début quant à l'utilisation des outils de prototypage rapide proposés. Plusieurs scénarios futurs peuvent être envisagés quant à l'usage de l'équipement et au développement du projet. Nous n'en citons ici que quelques uns.

Il serait intéressant d'implémenter la partie micro-ondes du système (incluant les antennes de réception et les modulateurs hautes fréquences). Ceci permettrait d'effectuer des mesures de canaux pour un tel système de communication à bande étroite. Ainsi, nous serions en mesure d'estimer le niveau d'interférences présentes selon divers environnements.

Un autre projet serait le développement de techniques de traitement de signal plus complexes que celles implémentées au courant de ce projet. La plateforme fournit beaucoup de capacité, notamment avec les quatre DSP très performants qui y sont souvent utilisés.

La plateforme fournit un bon potentiel pour le développement d'un système MIMO. Ici, nous serons en mesure d'implémenter un système MIMO  $4 \times 4$  au plus. L'utilisation de Simulink permettra au concepteur débutant d'avoir une bonne vue de haut niveau du système. Par ailleurs, un concepteur expérimenté peut toujours implémenter un système en passant par du codage VHDL et C/C++.

## RÉFÉRENCES

- [3GPP TS 25.223 v.8.1.0] 3rd Generation Partnership Project. Technical Specification Group Radio Access Network ; Spreading and Modulation (TDD) (Release 8)
- [3GPP TS 25.101 v.8.3.0] 3rd Generation Partnership Project. Technical Specification Group Radio Access Network ; User Equipment (UE) radio transmission and reception (FDD) (Release 8)
- [Alexander S. B, 1997] ALEXANDER, S. B. (1997), Receiver Noise Modeling. In *Optical Communication Receiver Design*. IET. ISBN : 978-0-85296-900-7.
- [Barbero, L.G. & Thompson, J.S., 2005] Barbero, L.G. & Thompson, J.S. (1999). Rapid Prototyping System for the Evaluation of MIMO Receive Algorithms. *The International Conference on Computer as a Tool*, Volume 2, Pages 1779-1782.
- [Crouch L. W, 2001] Crouch, L. W. (2001). Frame Synchronization. In *Digital and Analog Communication Systems* (Sixth Edition, P. 200-202). Upper Saddle River, New Jersey : Prentice-Hall, Inc. ISBN : 0-13-08-1223-4.
- [Crystals1] <http://www.radio-electronics.com/info/data/crystals/ocxo.php>
- [Crystals2] <http://www.citizencrystal.com/images/pdf/k-cm315.pdf>
- [Crystals3] <http://www.ctscorp.com/components/Datasheets/1914.pdf>
- [Fortin A.,2001] Fortin A. (2001). *Analyse Numérique pour Ingénieurs*. Montreal, Canada : Presses Internationales Polytechnique. ISBN : 2-553-00936-4.
- [Frigon J.-F. & Daneshrad, B., 1999] Frigon J.-F. & Daneshrad, B. (1999). Field measurements of high speed QAM wireless transmission using equalization and real-time beamforming. *Global Telecommunications Conference*, Volume 4, Pages 2102-2106.
- [Frigon J.-F. & Daneshrad, B., 2002] Frigon J.-F. & Daneshrad, B. (Jan. 2002). Field measurements of an indoor high-speed QAM wireless system using decision feed-



back equalization and smart antenna array. *IEEE Transactions on Wireless Communications*, Volume 1, Issue 1, Pages 134-144.

[Goldsmith A., 2005] Goldsmith, A. (2005). *Wireless Communications*. New York, NY : Cambridge University Press. ISBN-13 : 978-0-512-83716-3.

[Haccoun D., 2006] HACCOUN, D., Notes de Cours ELE6701A, Détection, décision et estimation des signaux, Août 2006.

[Hashemi, H., 1993] Hashemi, H.(1993). The Indoor Radio Propagation Channel. *Proceedings of the IEEE*, Volume 81, Issue 7, Pages 943-968.

[Lang, S. et al., 2004] Lang, S., Rao, M.R., Daneshrad, B. (June 2004). Design and development of a 5.25 GHz software defined wireless OFDM communication platform. *IEEE Communications Magazine*, Volume 42, Issue 6, Pages 6-12

[Lyrtech DRC V4] Lyrtech Inc. (October 2007). DRC Module User's guide.

[Lyrtech FastBus] Lyrtech Inc. (October 2007). FastBus Technical Reference Guide.

[Lyrtech SDRAM] Lyrtech Inc. (December 2007). FPGA SDRAM Record and Playback Technical Reference Guide.

[Lyrtech SMQ DesignGuide] Lyrtech Inc. (December 2007). SignalMaster Quad Virtex-4 Model-based Design Guide.

[Lyrtech SMQ UserGuide] Lyrtech Inc. (October 2007). VHS-ADC/DAC User's Guide.

[Lyrtech VHS DesignGuide] Lyrtech Inc. (December 2007). VHS-ADC/DAC Model-based Design Guide.

[Lyrtech VHS UserGuide] Lyrtech Inc. (October 2007). SignalMaster Quad Virtex-4 User's Guide.

[Meyr, H., Ascheid, G., 1997] Meyr, H., ASCHEID, G. (1997). *Synchronization in Digital Communications* (Volume 1, Edition 1). John Wiley & sons. ISBN-13 : 978-0-471-50275-3

[Mitola, J., 1995] Mitola, J. (1995). The Software Radio Architecture. *IEEE Communications Magazine*, Volume 33, Pages 26-38.

- [Mueller, K.H. & Muller, M., 1976] Mueller, K.H. & Muller, M. (May 1976). Timing Recovery in Digital Synchronous Data Receivers. *IEEE Transactions on Communications*, Volume 24, Issue 5, Pages 516-531.
- [Nakajima, N. et al, 2001] Nakajima, N., Kohno, R., & Kubota S. (August 2001). Research and Developments of Software-Defined Radio Technologies in Japan. *IEEE Communications Magazine*, Pages 146-155.
- [Pahlavan K., 1988] Pahlavan, K. (1988), Chapter 22 - Signal Processing in Telecommunications. In *Signal Processing Handbook*, CHEN C.H.(ed.)(Pages 683-722). CRC Press. ISBN : 978-0-824-77956-6.
- [Proakis, J., 2000] Proakis, J. (2000). *Digital Communications, Second Edition* (4th edition). McGraw-Hill. ISBN-13 : 978-0-072-32111-1.
- [Qureshi, 1985] Qureshi, S. U. H. (Sept. 1985). Adaptive Equalization. *Proceedings of IEEE*, Volume 73, Issue 9, Pages 1349-1387.
- [Raghu M.R. et al, 2004] Raghu et al. (2004). Multi-Antenna Testbeds for Research and Education in Wireless Communications. *IEEE Communications Magazine*, Volume 42, Issue 12, Pages 72-81.
- [Rappaport, T.S., 2002] Rappaport, T.S. (2002) *Wireless Communications Principles and Practice* (2nd Edition). Prentice Hall. ISBN-13 : 978-0-130-42232-3.
- [Roy, S. et al, 2008] Roy, S., Boudrealt, J-F., Dupont, L. (2008). An end-to-end prototyping framework for compliant wireless LAN transceivers with smart antennas. *Computer Communications*, Volume 31, Pages 1551-1563.
- [Safadi M.S. & Ndzi D.L., 2006] Safadi M.S. & Ndzi D.L.(2006) Digital Hardware Choices For Software Radio (SDR) Baseband Implementation. *Information and Communications Technologies, ICTTA*, Volume 2, Pages 2623-2628.
- [Saleh, A. & Valenzuela, R., 1987] Saleh, A. A. M. & Valenzuela (Feb. 1987), R. A., A Statistical Model for Indoor Multipath Propagation. *IEEE Journal on Selected Areas in Communications*, Volume 5, Issue 2, Pages 138-137.

- [SDR Forum] <http://www.sdrforum.org/pages/aboutSdrTech/whatIsSdr.asp>
- [Sklar, B., 2001] Sklar, B. (2001). *Digital Communications : Fundamentals and Practice* (2nd Edition). Prentice Hall PTR. ISBN-13 : 978-0-130-84788-1.
- [Stephens, D.R., 2001] STEPHENS, D. R. (2001). *Phase-Locked Loops for Wireless Communications, Digital, Analog and Optical Implementations* (Second Edition). Springer. ISBN-13 : 978-0-792-37602-6.
- [Stuber G.L. et al, 2004] Stuber G. L., Barry J. R., McLaughlin S. W., M. A. Ingram Y. Li, & Pratt T. G. (2004). Broadband MIMO-OFDM Wireless Communications. *Proc. of the IEEE*, Feb. 2004, vol. 92, pp. 271-294.
- [Takai, M. et al, 2005] Takai, M., Bagrodia, R., Gerla, M., Daneshrad, B., Fitz, M., Belding-Royer, M.S.E., Krishnamurthy, S. ; Molle, M., Mohapatra, P., Rao, R., Mitra, U., Shen, C.-C., Evans, J. (Feb. 2005). Scalable testbed for next-generation wireless networking technologies. *First International Conference on Tridentcom 2005*, Feb 2005, Pages 162-171
- [Terr] Terr, D. & Weisstein, E. W. *Barker Code*. MathWorld, A Wolfram Web Resource. Tiré de <http://mathworld.wolfram.com/BarkerCode.html>
- [Tse D. & Viswanath, P., 2005] Tse, D. & Viswanath, P. (2005). *Fundamentals of Wireless Communication*. New York, USA : Cambridge University Press. ISBN-13 : 978-0-521-84527-4.
- [Wakerly, J.F, 2001] Wakerly J.F. (2001). *Digital Design : principles and practices* (Third Edition). Upper Saddle River, New Jersey : Prentice-Hall, Inc. ISBN : 0-13-089896-1
- [WHYNET] <http://chenyen.cs.ucla.edu/projects/whynet/index.html>
- [Wozencraft, J.M & Jacobs, I.M., 1965] Wozencraft, J.M & Jacobs, I.M. (1965). *Principles of Communication Engineering*, Wiley, New York.
- [Widrow, B., 1985] Widrow, B. & Stearns, S. (1985). *Adaptive Signal Processing*. Prentice Hall. ISBN : 0130040290

[Ziemer, R.E & Peterson R.L., 2000] ZIEMER, R. E. & PETERSON, R. L. (2000). *Introduction to Digital Communications* (Second Edition). Prentice Hall PTR. ISBN-13 : 978-0-138-96481-8.

## ANNEXE I

### QUELQUES BASES THÉORIQUES

#### I.1 Les récepteurs à maximum de vraisemblance [Proakis, J., 2000]

##### I.1.1 Structure du récepteur

Soit une séquence binaire  $m_i$  constituée de  $K$  bits tel que  $m_i = \{b_1, \dots, b_K\} \in \nu$  où  $\nu$  est la série de tous les  $M = 2^K$  messages possibles. Chaque message peut être encodé selon un signal analog  $s_i(t)$  défini sur un interval temporel  $[0, T)$ . S'un point de vue minimisation du taux d'erreurs binaire, un récepteur optimal effectue une corrélation du signal de réception avec chacun des signaux de transmission possible suivi d'un échantillonnage à une période  $T$  (voir figure I.1). Le signal transmis est estimé en se basant sur la distance de la valeur de la corrélation avec les différentes valeurs possibles.

À noter que qu'un récepteur à corrélation est équivalent à un filtre adapté du point de vue de la réponse à l'instant d'échantillonnage  $T$  [Haccoun D., 2006].

##### I.1.2 Correspondance dans un espace vectoriel

Grâce au processus d'orthogonalisation de Gram-Schmidt, les signaux  $\{s_1(t), \dots, s_M(t)\}$  peuvent être représentés comme une combinaison linéaire de  $N \leq M$  fonctions de base orthonormaux,  $\{\phi_1, \dots, \phi_N\}$ . Ainsi chacun des signaux formera un point d'une constellation de  $M$  points dans un espace à  $N$  dimensions. Pour un signal  $s_i(t)$ , les coordonnées

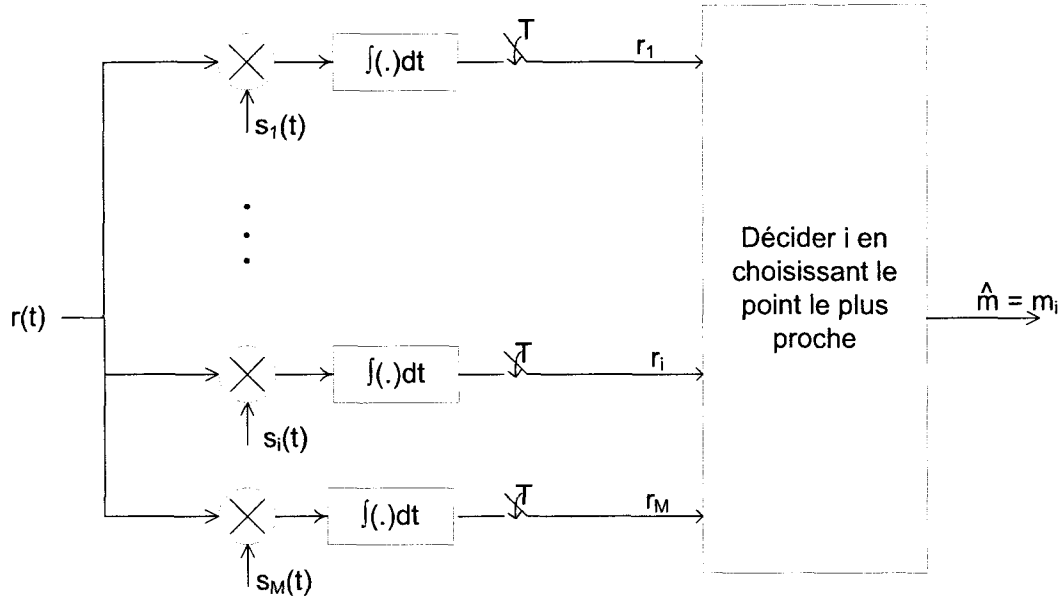


FIGURE I.1 Récepteur à corrélation

du point correspondant sera alors obtenu par la série d'équations I.1.

$$s_{ij} = \int_0^T s_i(t) \phi_j(t) dt, \text{ pour } j = 1 \dots N \quad (\text{I.1})$$

Le récepteur doit décider du signal transmis à partir du signal de réception  $r(t)$ . Le  $j^{ime}$  coefficient du vecteur de réception est obtenu à partir de la corrélation entre le  $r(t)$  et  $\phi_j(t)$ .

Ainsi, nous avons le développement suivant :

$$r_j = \int_0^T r(t) \phi_j(t) dt, \text{ pour } j = 1 \dots N \quad (\text{I.2})$$

Puisque l'ensemble de fonctions  $\phi_j(t)$  sont orthonormaux, par définition, nous avons :

$$\begin{aligned} \int_0^T \phi_j(t) \phi_k(t) dt &= 1, \text{ si } j = k \\ &= 0, \text{ sinon} \end{aligned} \quad (\text{I.3})$$

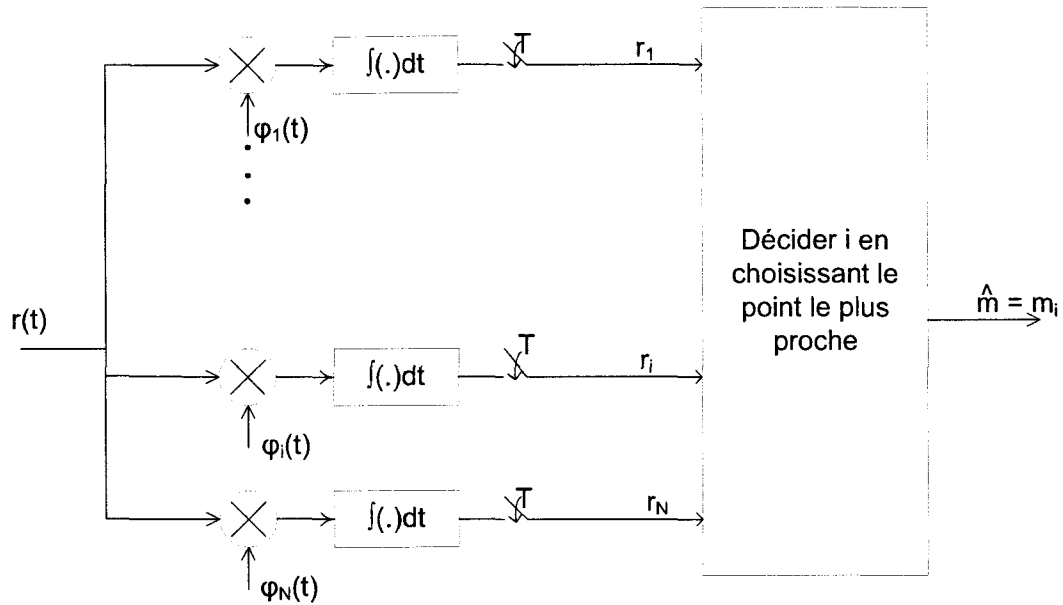


FIGURE I.2 Structure d'un récepteur pour la détection d'un signal dans le bruit AWGN

En posant l'hypothèse qu'un bruit additif Gaussien blanc  $n(t)$  corrompt la transmission,

$$\begin{aligned}
 r_j &= \int_0^T r(t) \phi_j(t) dt \\
 &= \int_0^T \sum_{n=1}^N (s_n + n_n) \phi_n(t) \phi_j(t) dt \\
 &= s_j + n_j
 \end{aligned} \tag{I.4}$$

Le message transmis peut donc être estimé en effectuant une projection dans l'espace orthonormal défini par  $\{\phi_1, \dots, \phi_N\}$  et décider selon le point de la constellation le plus proche de  $(r_1, \dots, r_N)$ . La figure I.2 illustre ce récepteur. Une telle structure permettra l'analyse de la transmission et de la réception dans un espace vectorielle. Ainsi, l'analyse de la performance pourra être simplifiée.

## I.2 Performances d'une modulation

Il existe plusieurs critères pour définir la performance d'une modulation digitale. Nous en faisons ici l'analyse de quelques uns dans le cas générale et dans le cas particulier d'une signalisation QPSK.

### I.2.1 Puissance de transmission et le SNR

Soit une constellation tel que les symboles sont transmises avec une puissance  $S_i$ . Chaque symbole a une certaine probabilité  $p_i$  d'être transmis. La puissance moyenne de transmission est alors donnée selon la formule I.5.

$$\bar{P} = \sum_{i=1}^M S_i p_i \quad (I.5)$$

Dans le cas d'une signalisation QPSK, tous les symboles sont équiprobables et d'énergie  $E_s$ . Alors la puissance moyenne de transmission est de  $E_s/T$ , où  $T$  est la durée d'un symbole. Par contre, pour une signalisation 8-QAM avec 4 symboles de puissance  $\frac{E_s}{2}$  et quatres autres de puissance  $\frac{3E_s}{2}$ , la puissance moyenne est alors de  $E_s$ .

Pour des raisons de coût d'implémentation, une des contraintes de conception est d'avoir un rapport de puissance crête à la puissance moyenne (PAPR) bas [Tse D. & Viswanath, P., 2005]. Dans cet optique, la constellation QPSK aura un PAPR de 1 alors que le PAPR du 8-QAM sera de 1,5.

En assumant que le signal occupe une largeur de bande passante  $W$  et que la densité spectrale de puissance du bruit est uniforme et égale à  $\sigma^2$ , nous pouvons en déduire l'équation



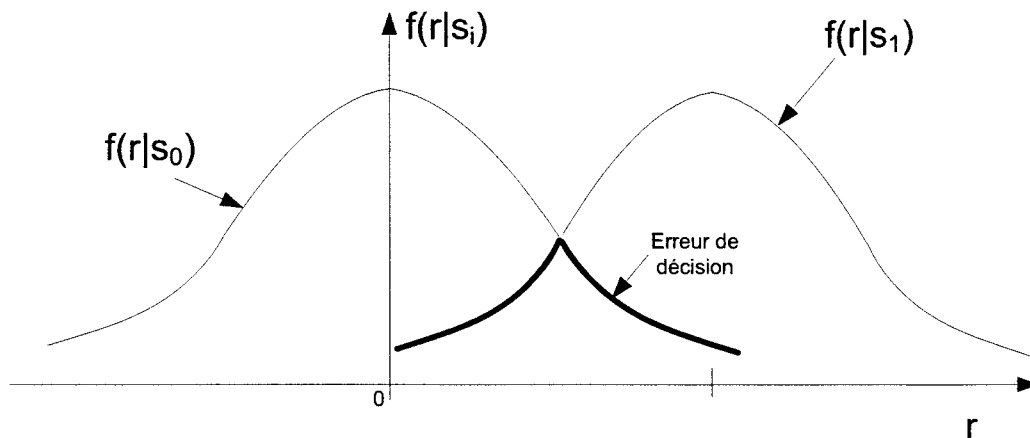


FIGURE I.3 Fonctions de densité de probabilité conditionnelles dans un canal AWGN

I.6 pour le rapport signal à bruit.

$$SNR = \frac{E_s}{\sigma^2 \cdot W} \quad (\text{I.6})$$

### I.2.2 Taux d'erreurs dans un canal AWGN

Considérons une signalisation où les signaux transmis dans un canal AWGN sont  $+d$  ou  $0$ . La distribution de densité de probabilité du signal de réception est illustrée à la figure I.3. Une erreur de décision est déclarée lorsque le symbole décidé est le complément de celui qui a été transmis. La région de l'erreur de décision est montrée par la ligne en gras.

Sans perte de généralité, nous pouvons affirmer que le bruit a une densité de puissance  $\frac{N_0}{2}$ . En utilisant la symétrie de la fonction Gaussienne par rapport à sa moyenne, nous

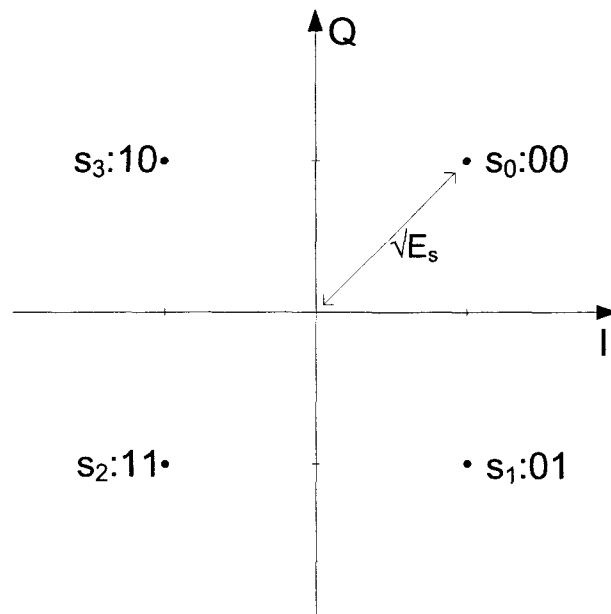


FIGURE I.4 Constellation QPSK

pouvons alors développer une expression pour la probabilité d'erreurs.

$$\begin{aligned}
 P(E) &= \frac{1}{2}P(E|S_0) + \frac{1}{2}P(E|S_1) \\
 &= Q\left(\frac{\frac{d}{2}}{\sqrt{\frac{N_0}{2}}}\right) \\
 &= Q\left(\frac{d}{\sqrt{2N_0}}\right)
 \end{aligned} \tag{I.7}$$

À partir de l'équation I.7, nous pouvons conclure que les chances d'avoir une erreur de décision augmentent lorsque la distance entre les deux points de la constellation diminue. Dans une situation où la constellation contient davantage de points, il faudra définir des espaces de décision, évaluer les probabilités d'erreurs paire à paire et en effectuer une moyenne selon les probabilités de transmission de chacun des symboles. Prenons, comme exemple, une constellation QPSK de puissance moyenne  $E_s$ .

Pour le symbole  $S_0$ , une bonne décision s'effectue si les parties réelles et imaginaires du

signal sont positives.

$$\begin{aligned}
 P(\text{Pas d'erreur}|S_0) &= P(r_1 > 0|S_0) \cdot P(r_2 > 0|S_0) \\
 &= \left[ 1 - Q\left(\frac{\sqrt{\frac{E_s}{2}}}{\sqrt{\frac{N_0}{2}}}\right) \right] \cdot \left[ 1 - Q\left(\frac{\sqrt{\frac{E_s}{2}}}{\sqrt{\frac{N_0}{2}}}\right) \right] \\
 &= \left[ 1 - Q\left(\sqrt{\frac{E_s}{N_0}}\right) \right]^2
 \end{aligned} \tag{I.8}$$

$$\begin{aligned}
 \Rightarrow P(E|S_0) &= 1 - P(\text{Pas d'erreur}|S_0) \\
 &= 1 - \left[ 1 - Q\left(\sqrt{\frac{E_s}{N_0}}\right) \right]^2 \\
 &\approx 2Q\left(\sqrt{\frac{E_s}{N_0}}\right)
 \end{aligned} \tag{I.9}$$

Donc,

$$\begin{aligned}
 P(E) &= \sum_{i=0}^3 P(S_i)P(E|S_i) \\
 &= 2Q\left(\sqrt{\frac{E_s}{N_0}}\right) \\
 &= 2Q(\sqrt{SNR})
 \end{aligned} \tag{I.10}$$

Lorsque nous utilisons un ordonnancement de Gray (voir figure I.4), le taux d'erreurs binaires est donné selon la formule I.11.

$$\begin{aligned}
 BER &= \frac{1}{2} \cdot 2Q(\sqrt{SNR}) \\
 &= Q(\sqrt{SNR})
 \end{aligned} \tag{I.11}$$

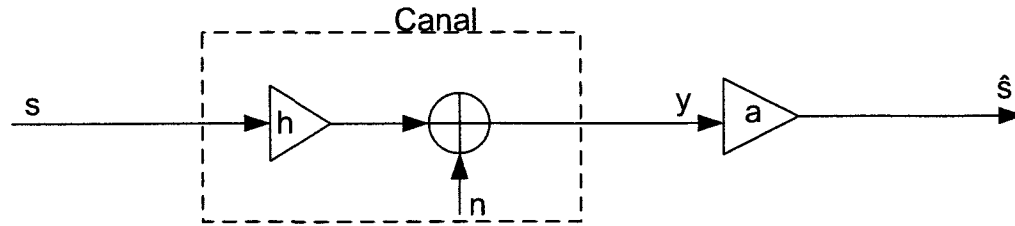


FIGURE I.5 Estimation linéaire d'une source à partir d'un signal de réception

### I.2.3 Efficacité spectrale

Nyquist a démontré que le taux maximal de transmission de symboles sans ISI est de  $R_s = 2W_c$  pour un canal de largeur de bande limité à  $W_c$ . Ainsi, le taux maximal de transmission de symboles par unité de largeur de bande est de  $2 \text{ symboles/s/Hz}$  [Sklar, B., 2001]. Afin de déterminer l'efficacité spectrale maximale théorique, il suffit de déterminer le nombre de bits représentés par symbole.

Dans une situation où 2 bits sont représentés par un symbole à un taux de  $R_s \text{ symboles/s}$ , l'efficacité spectrale maximale est de  $4 \text{ bits/s/Hz}$ . Une constellation QPSK utilise une largeur de bande de  $R_s \text{ Hz}$  pour transmettre  $2R_s \text{ bits/s}$  d'où une efficacité spectrale de  $2 \text{ bits/s/Hz}$ . Une signalisation 4-FSK transmettra aussi  $2R_s \text{ bits/s}$  mais requiert une largeur de bande de  $4R_s$ . Donc elle offre une efficacité spectrale de  $\frac{1}{2} \text{ bits/s/Hz}$ .

### I.3 Calcul de SNR à partir du MMSE

Cette démonstration est adaptée de l'exercice A8 tiré de [Tse D. & Viswanath, P., 2005]. Considérons un scalaire aléatoire  $x$  transmis à travers un canal d'évanouissement de gain  $h$ . Un bruit additif blanc Gaussien  $w$  corrompt le signal de réception  $y$ . Une estimation linéaire  $\hat{x} = ay$  est utilisée de sorte à minimiser l'erreur quadratique  $E[(x - \hat{x})^2]$ .

Nous pouvons utiliser le fait que  $x$  et  $n$  sont indépendants. Ceci permet le développement

de l'équation I.12.

$$\begin{aligned}
 \epsilon^2 &= E[(x - \hat{x})^2] \\
 &= E[(x^2 - 2x\hat{x} + \hat{x}^2)] \\
 &= E[(x^2 - 2(ahx^2 + anx) + (ahx + an)^2)] \\
 &= E[(x^2) - 2ahE[x^2] - 2aE[nx] + a^2h^2E[x^2] + 2a^2hE[xn] + a^2[n^2]] \\
 &= E[x^2] - 2ahE[x^2] + a^2h^2E[x^2] + a^2\sigma^2
 \end{aligned} \tag{I.12}$$

Pour trouver la valeur minimale de  $\epsilon^2$ , nous utilisons le fait que la dérivée par rapport à  $a$  doit être nulle. Ceci permet de trouver la valeur de  $a$  à l'équation I.15 de sorte que l'erreur quadratique soit minimale.

$$a = \frac{hE[x^2]}{h^2E[x^2] + \sigma^2} \tag{I.13}$$

Définissons le SNR à la réception tel qu'à l'équation I.14.

$$SNR = \frac{h^2E[x^2]}{\sigma^2} \tag{I.14}$$

En substituant I.15 et I.14 dans l'équation I.12, nous trouvons une expression pour le MMSE.

$$MMSE = \frac{E[x^2]}{1 + SNR} \tag{I.15}$$

Lorsque la puissance de transmission est unitaire ( $E[x^2] = 1$ ), nous pouvons exprimer le SNR en fonction du MMSE.

$$SNR = \frac{1}{MMSE} - 1 \tag{I.16}$$

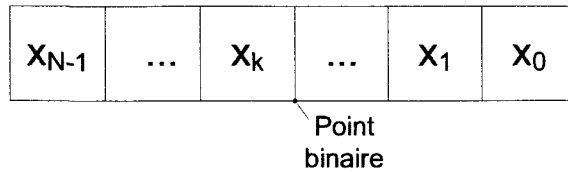


FIGURE I.6 Nombre en format binaire

#### I.4 Représentation de nombres binaires

[Fortin A.,2001] Il existe plusieurs techniques pour la représentation des nombres en format binaire. Les composantes FPGA et DSP des plateformes de développement utilisées fonctionnent en point fixe et nous avons utilisé une représentation complément à 2 pour les nombres signés. Un nombre en point fixe est défini par sa longueur (nombre de bits utilisés), la position du point binaire et le type de donnée (signé ou non-signé).

Soit un nombre représenté en format binaire complément à 2 selon la figure I.6. Nous montrons ci-dessous que ce nombre peut avoir différentes valeurs selon son interprétation.

N est le nombre de bits et k est la position du point binaire. Pour un nombre signé (Fix\_N\_k), la valeur en décimal est donnée par l'équation I.17.

$$\text{Valeur décimale} = -1 \cdot x_{N-1} + \sum_{i=0}^{N-2} x_i \cdot 2^{i-k} \quad (\text{I.17})$$

Si ce nombre doit être interprété comme non-signé (UFix\_N\_k), sa valeur en décimal est alors donnée par l'équation I.18.

$$\text{Valeur décimale} = \sum_{i=0}^{N-1} x_i \cdot 2^{i-k} \quad (\text{I.18})$$

Il existe d'autres types de représentation, notamment représentations en point flottant, que

nous traiterons pas ici.

## ANNEXE II

### DETAILS D'IMPLÉMENTATION SUR SIMULINK

#### II.1 Design du DSP

Les blocs nécessaires pour le design du DSP sont les suivants :

- Lyrtech SignalMasterQuad Virtex-4 Blockset \DSP\Onboard
  - DSP Board configuration
  - FastBus
- Lyrtech SignalMasterQuad Virtex-4 Blockset \DSP\Add-on hardware modules
  - DRC Control
- Lyrtech SignalMasterQuad Virtex-4 Blockset \DSP\Miscellaneous tools
  - Set bitstream
- Lyrtech SignalMasterQuad Virtex-4 Blockset \DSP\Remote control
  - DAC module control(Virtex-4)
  - VHS-ADC Virtex-4 control
- Lyrtech SignalMasterQuad Virtex-4 Blockset \Simulink\Shortcut tools
  - VHS FPGA Update Utility
- Signal Processing Blockset\Signal Processing Sources
  - Signal From Workspace

Ces blocs sont interconnectés tel qu'illustré à la figure II.1. Comme nous le constatons, la majorité des blocs utilisés ici effectuent du contrôle sur différents éléments des plateformes. Le *VHS FPGA Update Utility* n'est pas essentiel mais, comme nous le verrons, il est un raccourci pratique pour programmer le FPGA du VHS-ADC. Par la suite, nous fixons les paramètres suivants du modèle et de chacun des blocs.

- Simulation



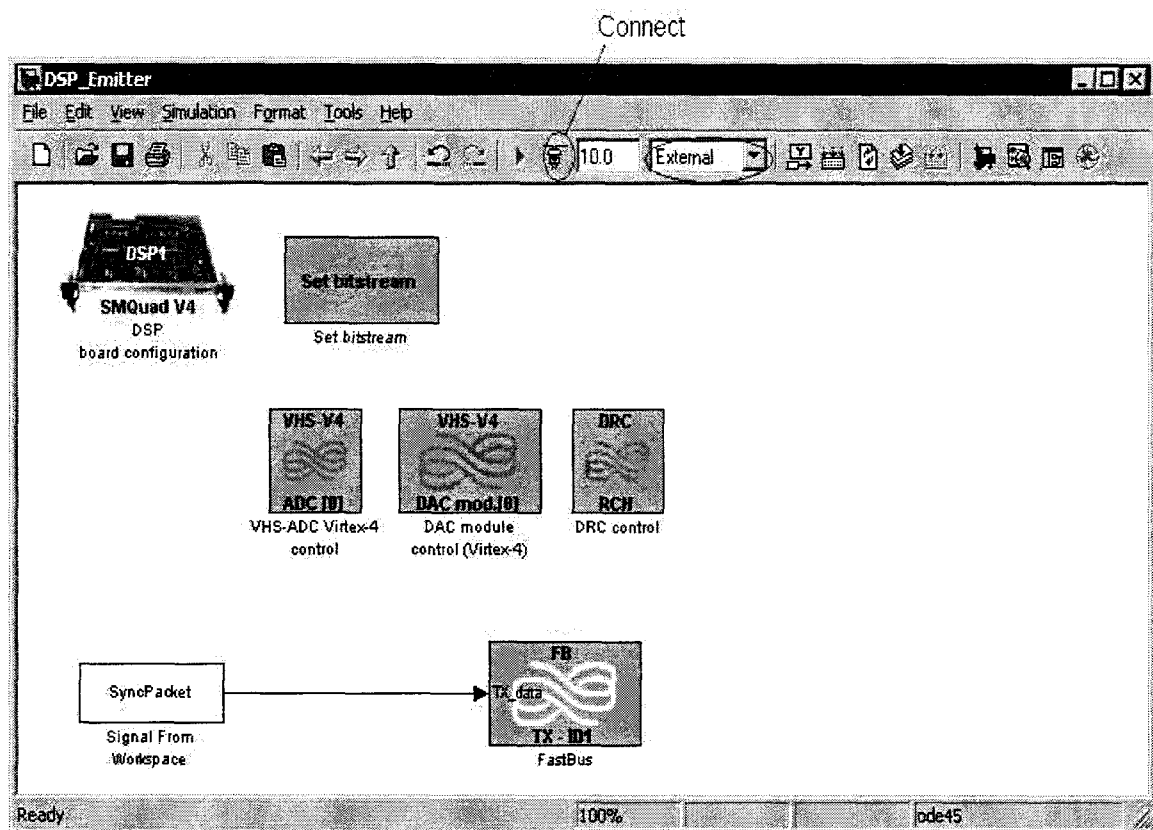


FIGURE II.1 DSP de l'émetteur

- Choisir l'option '*External*'. Cette action fournira le bouton 'connect' qui permettra par la suite de se connecter au DSP (voir figure II.1).
- Simulation \ Configuration Parameters
  - Solver > Solver Options
    1. *Type : Fixed-Step*
    2. *Solver : discrete (no continuous state)*
    3. *Tasking Mode for Periodic Sample Times : SingleTasking*
  - Real-Time Workshop > Target Selection
    1. À la variable *System Target File*, utiliser le bouton *Browse* pour choisir *rt\_smquadv4.tlc*.
    2. Real-Time Workshop\Interface > Data Exchange
    3. *Interface : External Mode*
    4. Dans *Host/Target Interface*, mettre *Transport Layer* à *External Communication for DSPLink*.
    5. Dans *Host/Target Interface*, mettre *MEX-File arguments* à '*autodetect*'.
- DSP Board Configuration
  - Onglet General Options
    1. *CPU Clock Frequency : 1000\_MHz*
    2. *Platform DSP ID : DSP 1*
  - Onglet Compiler Options
    1. Mettre *Optimization Level* à *-o3 :File* pour optimiser la grosseur du fichier généré.
  - Onglet Linker Options
    1. *Level 2 Memory Cache : L2\_Mode\_2\_(64KB)*
- FastBus
  - Onglet Basic
    1. *Channel ID : 1*

2. *Channel Direction* : TX
3. *Transfer Mechanism* : Event-Triggered DMA
4. *Subframe Length* : framesize/4
5. *Number of Subframes* : 4
6. *FPGA Data Type* : Unsigned
7. *FPGA Binary Point* : 0
8. *DSP Data Type* : uint32
9. *Sample Time* : 1

- DRC Control

- Onglet RapidCHANNEL

1. Cocher les cases *Use Front Panel RapidCHANNEL, Transmitter* et *Receiver*. Le paramètre *Clock Source* doit être à *Front panel received clock*
2. Cocher les cases *Use LYRIO+ RapidCHANNEL, Transmitter* et *Receiver*. Le paramètre *Clock Source* doit être à *LYRIO+ received clock*

- Onglet Advanced

1. *Explicit Sample Time* : framesize\*1000

- VHS-ADC Virtex-4 control

- Onglet General control

1. *First ADC cPCI slot ID* : slot [0]
2. *Number of controlled board(s)* : 1
3. *Clock source* : Fixed
4. Cocher la case *RUN*
5. *Sample time* : framesize \*1000

- Onglet Gain Control

1. *Gain control mode* : DSP [registers]

2. Cocher la case *Set all gain values as that of channel 1*
  3. *Channel 1 gain value* : 15
- DAC module control(Virtex-4)
    - Onglet General control
      1. Un message d'erreur apparaît mentionnant qu'un bloc de contrôle VHS-DAC doit être présent lorsque ce bloc est ajouté au modèle. L'utilisateur doit ignorer ce message et procéder à changer le paramètre suivant.
      2. *Carrier type* : VHS-ADC Virtex-4
    - Onglet Dual DAC 4
      1. *Operation Mode* : Full bypass
      2. *DAC channel A coarse gain* : 15
      3. *DAC channel B coarse gain* : 15
  - Signal From Workspace
    - *emphSignal* : Nom de la variable à transmettre dans le Workspace de Matlab. Nous avons utilisé la variable SyncPacket qui est un tableau d'entiers de dimension *frame-size*
    - *Sample time* : 1
    - *Samples per frame* : framesize
    - Mettre *Form output after final data value by* : à *Cyclic repetition*.

La compilation de ce modèle s'effectue à travers la séquence clavier *Ctrl + B*. Simulink fera alors appel à Code Composer Studio pour compiler le code généré par Real-Time Workshop et un exécutable de suffixe *.out* sera généré.

## II.2 Détails du FPGA du SMQUAD

Les blocs requis sont les suivants :

- Lyrtech SignalMasterQuad Virtex-4 Blockset \FPGA\Onboard
  - FPGA Board configuration
  - FastBus
  - RapidCHANNEL (2)
- Xilinx Blockset \Basic Elements
  - System Generator
  - Up Sample
- Simulink \Sinks
  - Terminator

Nous avons connecté les blocs tel qu'illustré à la figure II.2. Contrairement au DSP, System Generator de Xilinx requiert de fixer le champs *solver* à *variable-step* pour les paramètres de configuration (*Configuration Parameters*). Les paramètres de chacun des blocs sont les suivants :

- FPGA Board configuration
  - *Clock Type* : Free Running Hardware Clock
  - *Clock Source* : RapidCHANNEL - LYRIO
  - *Platform FPGA ID* : FPGA 1
  - Cocher la case *Copy bitstream to current directory after compilation*
- FastBus
  - *DSP ID* : 1
  - *Channel ID* : 1
  - *Direction* : RX
  - *Output Arithmetic Type* : Unsigned
  - *Output Width* : 32
  - *Binary Point* : 0
  - *Sample Period* : 1
- Up Sample

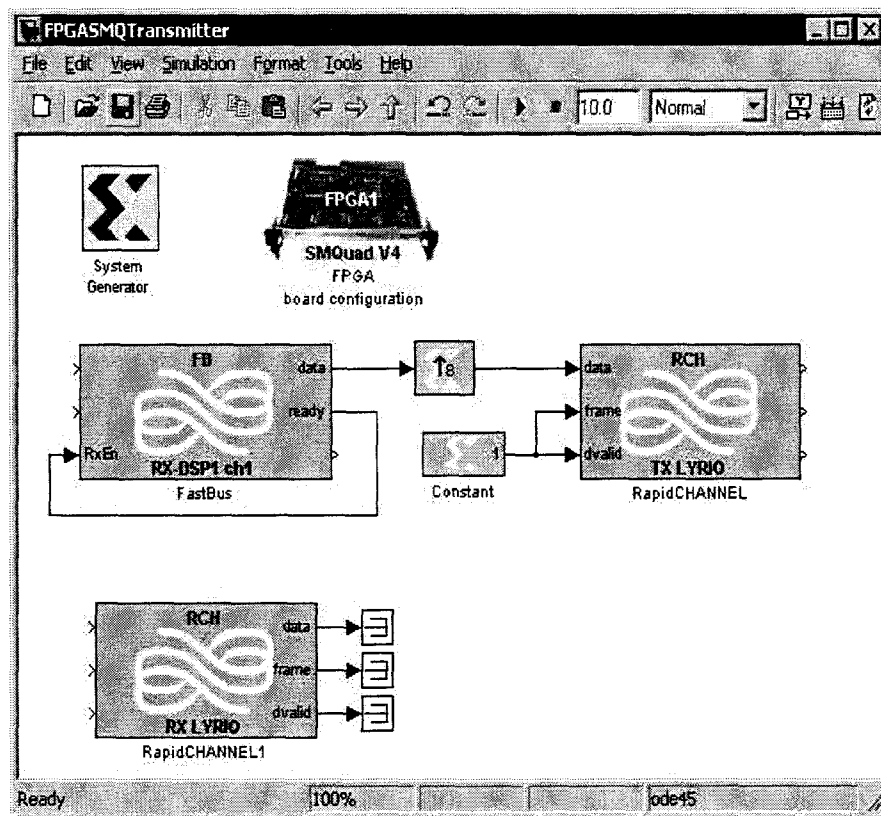


FIGURE II.2 FPGA sur le SMQuad au transmetteur

- *Sampling Rate* : 8
- Garder la case *copy samples* non-cochée.
- RapidCHANNEL
  - *Platform Type* : SMQUAD
  - *Interface* : LYRIO
  - *Direction* : TX
  - *Output Arithmetic Type* : Unsigned
  - *Output Width* : 32
  - *Binary Point* : 0
  - *Sample Period* : 1
- RapidCHANNEL1 (Pour la propagation de l'horloge du VHS-ADC)
  - *Platform Type* : SMQUAD
  - *Interface* : LYRIO
  - *Direction* : RX
  - *Output Arithmetic Type* : Unsigned
  - *Output Width* : 32
  - *Binary Point* : 0
  - *Sample Period* : 1
- System Generator
  - *Compilation* : Choisir *SMQUAD-V4* dans *Hardware Cosimulation\Lyrtech*
  - *Part* : Virtex4 xc4vlx100-10ff1148
  - *Synthesis Tool* : XST
  - *Target Directory* : *.netlistSMQuad*
  - *Hardware Description Language* : VHDL
  - *FPGA Clock Period (ns)* : 8. Cette valeur est utilisée dans le calcul des contraintes par XFLOW. Lyrtech suggère d'utiliser une valeur arrondie au minimum de 90% de la période de l'horloge du système. Ici, 8 ns correspond à la valeur arrondie à l'entier inférieur du 90% de 1/(104 MHz).

- *Simulink system period* : 1. Ceci correspond au minimum des temps d'échantillonnage pour les blocs du modèle.
- Cliquer sur *Generate* pour générer le bitstream nécessaire pour programmer le FPGA.

### II.3 Détails du VHS-ADC en transmission

Les blocs requis sont les suivants :

- Lyrtech VHS-ADC/DAC Virtex-4 Blockset \FPGA\Onboard
  - FPGA Board configuration
  - RapidCHANNEL (2)
- Lyrtech VHS-ADC/DAC Virtex-4 Blockset \FPGA\Add-on hardware modules
  - DAC module
- Xilinx Blockset \Basic Elements
  - System Generator
  - Cast (2)
  - Constant (3)
  - Slice (2)
  - Reinterpret (2)
- Xilinx Blockset \DSP
  - DAFIR v9\_0 (2)
  - FDA Tool
- Simulink \Sinks
  - Terminator

Après avoir connecté les blocs tel qu'illustré à la figure II.3, nous fixons les paramètres suivants pour chacun des blocs.

- FPGA Board configuration
  - *Clock Type* : Free Running Hardware Clock (Horloge dérivée des ADC)
  - *VHS-ADC/DAC Type* : VHS-ADC Virtex-4



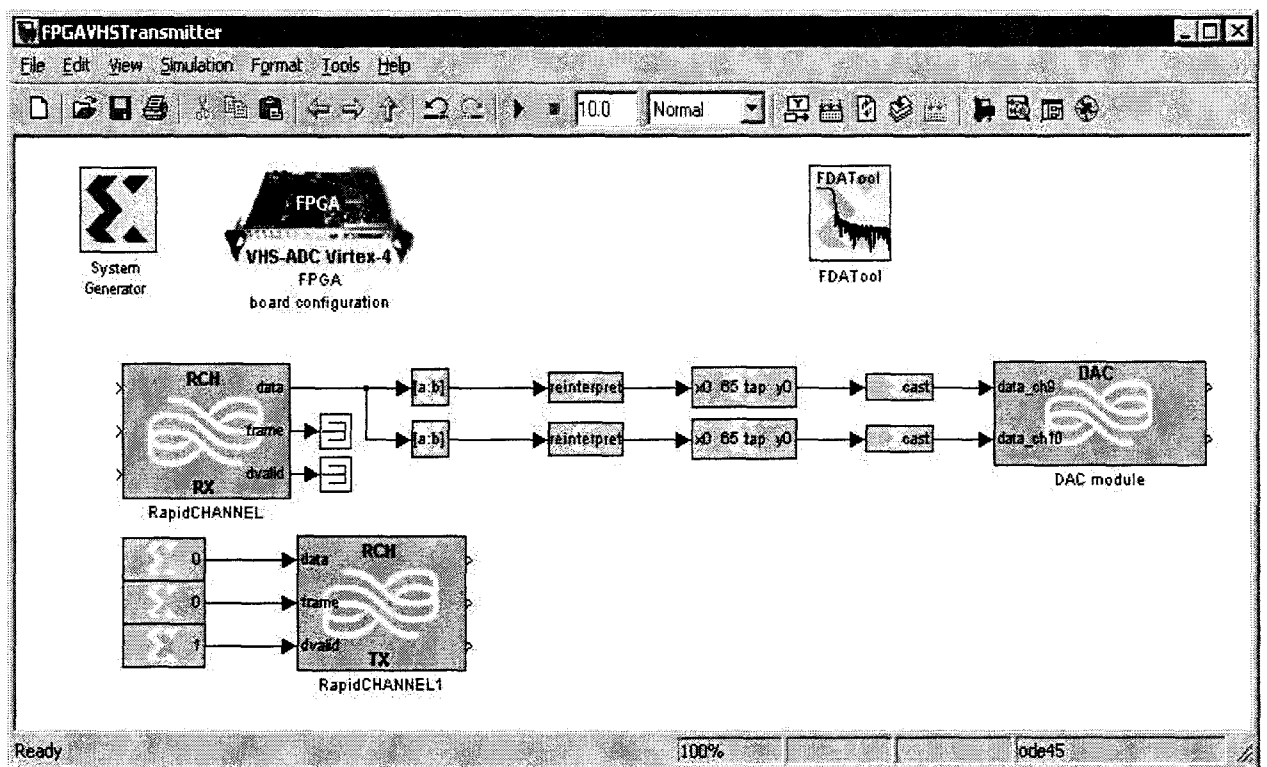


FIGURE II.3 FPGA sur le VHS du transmetteur

- Cocher la case Copy bitstream to current directory after compilation
- System Generator
  - *Compilation* : Choisir *VHS-ADAC-V4* dans *Hardware Cosimulation\Lyrtech*
  - *Part* : Virtex4 xc4vsx55-10ff1148
  - *Synthesis Tool* : XST
  - *Target Directory* : ./netlistVHS
  - *Hardware Description Language* : VHDL
  - *FPGA Clock Period (ns)* : 8
  - *Simulink system period* : 1
  - Cliquer sur *Generate* pour générer le bitstream nécessaire pour programmer le FPGA.
- RapidCHANNEL
  - *Platform Type* : VHS-ADC/DAC
  - *Direction* : RX
  - *Output Arithmetic Type* : Unsigned
  - *Output Width* : 32
  - *Binary Point* : 0
  - *Sample Period* : 1
- RapidCHANNEL1 (Pour la propagation de l'horloge vers la SMQUAD)
  - *Platform Type* : VHS-ADC/DAC
  - *Direction* : TX
  - *Output Arithmetic Type* : Unsigned
  - *Output Width* : 32
  - *Binary Point* : 0
  - *Sample Period* : 1
- Slice
  - *Width of Slice* : 14.
  - *Specify range as* : Sélectionner l'option *lower bit location + width*.
  - *Offset of bottom bit* : 14 pour le premier et 0 pour le deuxième.

- *Relative to* : Sélectionner l'option *LSB of input*.
- reinterpret
  - Cocher *Force Arithmetic Type* et sélectionner *Signed*.
  - Cocher *Force Binary Point*.
  - *Output Arithmetic Type* : Unsigned
  - *Output Binary Point* : 0
- cast
  - Choisir le type *Signed*.
  - *Number of bits* : 14 pour le premier et 0 pour le deuxième.
  - *Binary Point* : 0
  - *Quantization* : Choisir l'option *Truncate*.
  - *Overflow* : Choisir *Saturate*.
- FDATool
  - Dans la section *Response Type*, choisir *Raised Cosine*.
  - Dans la section *Design Method*, choisir *Window*.
  - Dans la section *Filter Order*, spécifier l'ordre du filtre à 64.
  - Dans la section *Frequency Specification*,
    1. Choisir *Normalized (0 to 1)*.
    2. Mettre  $w_c$  à 0,125.
    3. Mettre Roll-off à 0,22.
  - Dans la section *Magnitude Specification*, choisir *Square Root*
  - Pour générer le filtre, il suffit de cliquer sur le bouton *Design Filter*
- DAFIR v9\_0
  - *Coefficients* : `xlfsda_numerator('FDATool')`. Ceci indique que les coefficients proviennent du *FDATool*.
  - *Structure* : Inferred From Coefficients
  - *Number of bits* : 14
  - *Binary Point* : 14

- *Hardware Oversampling Rate* : 1
- *Latency* : 32
- DAC module
  - *Number of channels* : 2
  - *Binary Point* : 0
  - *Sample Period* : 1

Pour générer le bitstream, il faut cliquer sur le bouton *Generate* du bloc *System Generator*. Une fenêtre apparaît à la fin de la compilation pour la programmation du FPGA. Il suffit de choisir la plateforme appropriée et le bon bitstream (en général de la forme *vhs\_cw.bit*) et ensuite cliquer sur le bouton *Program FPGA*. Une fenêtre d'information apparaîtra lorsque le FPGA a fini d'être programmé.